Datenbanksysteme und bioinformatische Werkzeuge zur Optimierung biotechnologischer Prozesse mit Pilzen

Von der Fakultät für Lebenswissenschaften
der Technischen Universität Carolo-Wilhelmina
zu Braunschweig
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)
genehmigte
D i s s e r t a t i o n

von
Andreas Georg Grote
aus
Damme

Die vorliegende Arbeit wurde an der Technischen Universität Braunschweig, Fakultät für Lebenswissenschaften als Dissertation angefertigt.

Prüfungsvorsitzender: Professor Dr.-Ing. Dietmar C. Hempel
1. Referent: Professor Dr. Dieter Jahn
2. Referent: Professor Dr. Dietmar Schomburg

eingereicht am: 09.01.2008
mündliche Prüfung (Disputation) am: 08.05.2008

Druckjahr 2008

Danksagung

Diese Arbeit wäre ohne die Hilfe zahlreicher Menschen nicht zustande gekommen. Die wichtigsten Menschen möchte ich hier kurz erwähnen:

Mein erster Dank geht gleichermaßen an meine beiden Mentoren Professor Dr. Dietmar C. Hempel und Professor Dr. Dieter Jahn, die mir das Anfertigen dieser Arbeit ermöglicht haben. Beide standen mir gleichermaßen mit Rat und Tat bei Problemen zur Seite. Professor Dr. Dieter Jahn danke ich insbesondere dafür, dass er als Erstgutachter diese Arbeit bewertet. Professor Dr. Dietmar C. Hempel danke ich außerdem dafür, dass er den Vorsitz der Promotionskommission übernommen hat.

Ich danke Professor Dr. Dietmar Schomburg für die Erstellung des Zweitgutachtens dieser Arbeit.

Bei Dr. Karsten Hiller möchte ich mich ganz besonders bedanken, da er mir durch seine Hilfe bei technischen Problemen stundenlanges Suchen im Internet erspart hat. Mindestens genauso wertvoll waren seine fachlichen Ratschläge und seine Diskussionsbereitschaft, durch die ich sehr viele Anregungen gewonnen habe.

Dr. Richard Münch möchte ich ebenfalls für fachliche Ratschläge und seine Diskussionsbereitschaft danken. Ohne seine Hilfe wäre diese Arbeit nicht möglich gewesen.

Meinen Kollegen im Teilprojekt B4 danke ich, dass sie durch wertvolle Hinweise eine Verbesserung der *Aspergillus*-Datenbank an Benutzeranforderungen ermöglichten. Ganz besonders möchte ich in diesem Zusammenhang Alex Dalpiaz, Guido Melzer und Martin Kucklick nennen. Ein besonderer Dank gebührt ebenfalls dem Teilprojektleiter Dr. Bernd Nörtemann.

Bei Isam Haddad und Johannes Klein möchte ich mich bedanken, dass sie mir, besonders bei der Erstellung dieser Arbeit in LaTeX, hilfreich zur Seite standen.

Allen Mitarbeitern und ehemaligen Mitarbeitern der Bioinformatik am Institut für Mikrobiologie möchte ich für eine fruchtbare Zusammenarbeit danken. Die freundschaftlich-kollegiale Atmosphäre war immer so, dass ich mich sehr wohl gefühlt

habe. Besonders möchte ich hier erwähnen: Boyke Bunk, Claudia Pommerenke, Dr. Ida Retter und Maurice Scheer.

Allen Mitarbeitern im Arbeitskreis Jahn und am Institut für Bioverfahrenstechnik danke ich für eine freundliche und gute Atmosphäre während der Promotionszeit.

Nicht zuletzt möchte ich mich bei meinen Eltern und meinen Geschwistern bedanken, die eine wichtige Stütze in meinem Leben bilden.

Vorveröffentlichungen der Dissertation

Teilergebnisse aus dieser Arbeit wurden mit Genehmigung der Fakultät für Lebenswissenschaften, vertreten durch den Mentor der Arbeit, in folgenden Beiträgen vorab veröffentlicht:

Publikationen

Choi, C., Münch, R., Leupold, S., Klein, J., Siegel, I., Thielen, B., Benkert, B., Kucklick, M., Schobert, M., Barthelmes, J., Ebeling, C., Haddad, I., Scheer, M., Grote, A., Hiller, K., Bunk, B., Schreiber, K., Retter, I., Schomburg, D. & Jahn, D. (2007). SYSTOMONAS - an integrated database for systems biology analysis of *Pseudomonas*. *Nucleic Acids Res,* **35** (Database issue), D533–D537.

Grote, A., Hiller, K., Scheer, M., Münch, R., Nörtemann, B., Hempel, D. C. & Jahn, D. (2005). JCat: a novel tool to adapt codon usage of a target gene to its potential expression host. *Nucleic Acids Res,* **33** (Web Server issue), W526–W531.

Hiller, K., Grote, A., Maneck, M., Münch, R. & Jahn, D. (2006). JVirGel 2.0: computational prediction of proteomes separated via two-dimensional gel electrophoresis under consideration of membrane and secreted proteins. *Bioinformatics,* **22** (19), 2441–2443.

Hiller, K., Grote, A., Scheer, M., Münch, R. & Jahn, D. (2004). PrediSi: prediction of signal peptides and their cleavage positions. *Nucleic Acids Res,* **32** (Web Server issue), W375–W379.

Melzer, G., Dalpiaz, A., Grote, A., Kucklick, M., Göcke, Y., Jonas, R., Dersch, P., Franco-Lara, E., Nörtemann, B. & Hempel, D. C. (2007). Metabolic flux analysis using stoichiometric models for *Aspergillus niger*: Comparison under glucoamylase-producing and non-producing conditions. *J Biotechnol,* **132** (4), 405–417.

Münch, R., Hiller, K., Grote, A., Scheer, M., Klein, J., Schobert, M. & Jahn, D.

(2005). Virtual Footprint and PRODORIC: an integrative framework for regulon prediction in prokaryotes. *Bioinformatics,* **21** (22), 4187–4189.

Scheer, M., Klawonn, F., Münch, R., Grote, A., Hiller, K., Choi, C., Koch, I., Schobert, M., Härtig, E., Klages, U. & Jahn, D. (2006). JProGO: a novel tool for the functional interpretation of prokaryotic microarray data using Gene Ontology information. *Nucleic Acids Res,* **34** (Web Server issue), W510–W515.

Yang, Y., Malten, M., Grote, A., Jahn, D. & Deckwer, W.-D. (2007). Codon optimized *Thermobifida fusca* hydrolase secreted by *Bacillus megaterium*. *Biotechnol Bioeng,* **96** (4), 780–794.

Tagungsbeiträge

Dalpiaz, A., Grote, A., Kucklick, M., Lu, X., Melzer, G., Wurm, M., Rinas, U., Nörtemann, B., Dersch, P., Jahn, D. & Hempel, D. C. (2006). Systems biology of product and pellet formation in *Aspergillus niger*. Vereinigung für Allgemeine und Angewandte Mikrobiologie (VAAM) - Jahrestagung, Jena, Germany. Poster.

Grote, A., Dalpiaz, A., Melzer, G., Dersch, P., Nörtemann, B., Hempel, D. C. & Jahn, D. (2006). Investigations of the systems biology of *Aspergillus niger*. European Conference on Computational Biology (ECCB), Eillat, Israel. Poster.

Grote, A., Hiller, K., Scheer, M., Münch, R., Nörtemann, B., Hempel, D. C. & Jahn, D. (2005). JCat: A novel tool to adapt codon usage of a target gene to its potential expression host. German Conference on Bioinformatics (GCB), Hamburg, Germany. Poster.

Grote, A., Scheer, M., Hiller, K., Jahn, D. & Münch, R. (2004). PRODORIC: A Comprehensive Database on Cell Regulation in Prokaroytes. European Conference on Computational Biology (ECCB), Glasgow, United Kingdom. Poster.

Weitere

Lizensierung der Software „JCat - Java Codon Adaptation Tool“ für die Firma Sanofi-Aventis Deutschland GmbH in 2007.

Inhaltsverzeichnis

1 Einleitung

1.1 Bioinformatik

Die Bioinformatik ist ein sich rasant entwickelndes Teilgebiet der Biologie. In ihr vereinen sich sowohl Methoden der Mathematik, als auch der elektronischen Datenverarbeitung, um biologische Fragestellungen zu lösen. Zu den Themengebieten der Bioinformatik gehören die Sequenzanalyse von DNA und Proteinen, die Genomannotation, die Analyse von Genexpressionsdaten, die Beschreibung von metabolischen und Regulationsnetzwerken, vergleichende Genomanalysen und die Vorhersage von Proteinstrukturen. Zur Bearbeitung der spezifischen Fragestellung

Tabelle 1.1: Spezialisierte biologische Datenbanken.

Datenbank	Schwerpunkt	Internetadresse
BRENDA	Enzymdaten	`http://www.brenda-enzymes.info`
EMBL	Nukleotidsequenzen	`http://www.ebi.ac.uk/embl`
KEGG	Metabolische Daten	`http://www.genome.jp/kegg`
PDB	Proteinstrukturen	`http://www.pdb.org`
Prodoric	Transkriptionsfaktoren und regulatorische Daten von Prokaryonten	`http://www.prodoric.de`
Swiss-Prot	Proteinsequenzen	`http://expasy.org/sprot`

stehen eine Vielzahl von Techniken und Konzepten der angewandten Mathematik und Informatik zur Verfügung. So werden beispielsweise „Hidden Markov Modelle (HMM)“ in der Genomannotation eingesetzt um kodierende Bereiche von nicht kodierenden Bereichen der DNA zu unterscheiden [53]. „Supported Vector Machines (SVM)“ werden verwendet um Microarraydaten zu analysieren [26]. Auch Algorithmen der künstlichen Intelligenz wie zum Beispiel „künstliche neuronale Netze (KNN)“ finden in der Bioinformatik Verwendung, um zum Beispiel Promotorsequenzen von bakteriellen Genomen vorherzusagen [40]. Algorithmen sind zum

Teil für konkrete Fragestellungen der Biologie entwickelt worden. So versteht man unter dem Prozess der Alignmenterstellung das Arrangieren von DNA, RNA oder Proteinsequenzen in der Art, dass verwandtschaftliche, strukturelle oder funktionelle Beziehungen zwischen den einzelnen Sequenzen abgeleitet werden können. Man unterscheidet zwischen globalen und lokalen, sowie zwischen paarweisen und multiplen Sequenzalignments. Die wichtigsten Algorithmen, die für dieses komplexe Problem entwickelt wurden sind: Needleman-Wunsch-Algorithmus (globales Alignment) [57], Smith-Waterman-Algorithmus (lokales Alignment) [70], FASTA (heuristisches Verfahren für ein paarweises multiples Alignment) [49] und BLAST (Sammlung von Programmen für heuristische paarweise multiple Alignments) [2]. Neben der Gewinnung neuer Information aus experimentellen Daten, stellt die Speicherung eben dieser experimentellen Daten und der gewonnenen Informationen einen weiteren wichtigen Bereich der Bioinformatik da. Die explosionsartige Entwicklung des Internets hat gerade in dieser Disziplin entscheidenden Einfluss gehabt. Techniken zur Speicherung und Darstellung von Informationen stehen dadurch in einer breiten Fülle zur Verfügung. So werden die Daten in Datenbanken gespeichert und mittels Internet anderen Wissenschaftlern zur Verfügung gestellt. Die meisten biologischen Datenbanken sind dabei auf spezifische Fragestellungen

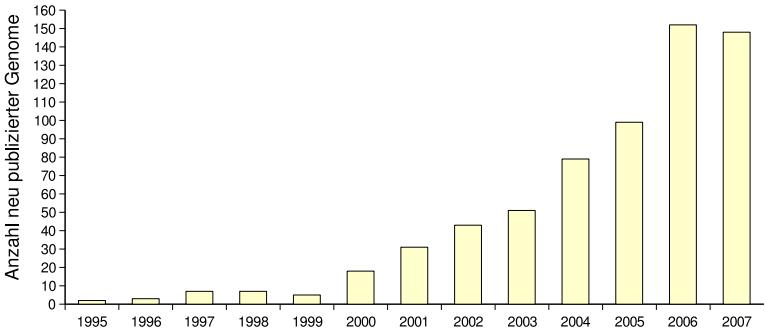

Abbildung 1.1: Dargestellt ist die Anzahl neu publizierter Genome pro Jahr für die letzten 12 Jahre (Quelle: `http://www.genomesonline.org`).

spezialisiert. In Tabelle 1.1 sind einige wichtige Datenbanken mit dem jeweiligen Schwerpunkt angegeben.

Die Grundlage vieler biologischer Datenbanken bilden Nukleotidsequenzen aus Genomprojekten. Die Anzahl publizierter Genome nimmt stetig zu. In Abbildung 1.1 ist diese Entwicklung graphisch dargestellt. Da die Kosten für die Sequenzierung kompletter Genome immer niedriger werden, wird dieser Trend vermutlich noch an Geschwindigkeit gewinnen. Aufgabe der Bioinformatik ist es hier geeignete Werkzeuge zur Verfügung zu stellen, die eine präzise Annotation der neuen Genome ermöglichen und die Ergebnisse Benutzern zugänglich zu machen.

1.2 Systembiologie

Die Systembiologie ist eng mit der Bioinformatik verbunden, da die Systembiologie, mehr als andere Zweige der Biologie, auf Werkzeuge der Bioinformatik angewiesen ist, um neue Gesetzmäßigkeiten zu entdecken. Ziel der Systembiologie ist es einen Organismus in seiner Gesamtheit zu verstehen und alle untersuchbaren Ebenen in ein Modell zu integrieren, das Vorhersagen über biologische Abläufe erlaubt. Eine herausragende Rolle für die Systembiologie spielen dabei die sogenannten ‚Omics'-Techniken, die eine Analyse der Gesamtheit einer Regulationsebene eines Organismus oder einer Zelle ermöglichen [45]. Für die Transkriptionsebene ermöglichen „Microarrays" einen Blick auf die RNA, die in der betrachteten Zelle zu einem bestimmten Zeitpunkt vorhanden ist. Die Proteomebene wird beispielsweise mit Hilfe von „Zwei-Dimensionalen-Protein-Gelen" oder „Massenspektrometrie" untersucht. Für die Analyse der Metabolite (Metabolomics) in einer Zelle stehen zum Beispiel die „Gaschromtatografie mit Massenspektrometrie (GC/MS)" und „Flüssigchromatographie mit Massenspektrometrie (LC/MS)" zur Verfügung.

Bei der Verarbeitung der dabei entstandenen Datenmengen sind geeignete Algorithmen und Computerprogramme, die beispielsweise Transkriptomdaten gruppieren [7], Unterschiede bei Proteomdaten visualisieren [10] oder Metabolomdaten [43] auswerten, ein unentbehrliches Hilfsmittel geworden.

Ein wichtiges Ziel der Systembiologie ist die Bildung von Modellen für komple-

xe biologische Prozesse. Für die einzelnen Ebenen stehen dafür verschiedene Konzepte zur Verfügung. Das Konzept der „Flux balance analysis“ zum Beispiel kann eingesetzt werden, um den Metabolismus eines Bakteriums quantitativ zu simulieren [42]. Für Transkriptomanalysen stehen verschiedene „Clusteralgorithmen“ zur Verfügung, die Gene nach gleichen Expressionsprofilen einteilen und so Hinweise darauf liefern können, welche Gene einem gemeinsamen Regulon angehören [23].

Die Integration der einzelnen Regulationsebenen zu einem Modell der Zelle oder des Organismus ist die Herausforderung, der sich die Systembiologie in den nächsten Jahren stellen muss [54].

1.3 Der Sonderforschungsbereich 578: Vom Gen zum Produkt

Der Sonderforschungsbereich 578 ist ein Zusammenschluss verschiedener Institute der Technischen Universität Braunschweig, dem Helmholtz Zentrum für Infektionsforschung und dem Max-Planck-Institut Magdeburg. Der Schwerpunkt des SFB 578 liegt auf der Erforschung der rekombinanten Produktion biotechnologisch oder pharmazeutisch interessanter Proteine mittels gentechnisch veränderter Organismen. Als Wirtssystem für die rekombinante Proteinproduktion dienen dabei *Bacillus megaterium* als Gram positives prokaryotisches System und *Aspergillus niger* als eukaryotisches System.

Innerhalb des SFB werden Forschungsansätze aus verschiedenen wissenschaftlichen Disziplinen gebündelt, um ein möglichst umfassendes Modell des Produktionsprozesses zu erhalten. Der Produktionsprozess soll somit in seiner Gesamtheit verstanden werden und so optimiert werden. Die Forschungsansätze kommen aus den Bereichen der Molekularbiologie, der Mikrobiologie, der technischen Chemie, der Biotechnologie, der pharmazeutischen Technologie, der Bioverfahrenstechnik, der mechanischen, chemischen und thermischen Verfahrenstechnik, der Bioinformatik, der elektrischen Messtechnik und der Mikrotechnik. Um dem interdisziplinären Forschungsansatz gerecht zu werden, unterteilt sich der SFB in vier Projektbereiche, die sich weiter in insgesamt 17 Teilprojekte aufschlüsseln. Die vier Projektbereiche sind Projektbereich A: Molekularbiologie der Produktbildung, Projektbe-

reich B: Systembiotechnologie der Produktbildung, Projektbereich C: Prozesstechnik und Projektbereich D: Anwendungstechnik [32].

1.3.1 Das Teilprojekt B4: Systembiologie der Produkt- und Pelletbildung durch *Aspergillus niger*

Das Teilprojekt B4 mit dem Titel „Systembiologie der Produkt- und Pelletbildung durch *Aspergillus niger*“ beschäftigt sich mit der Herstellung rekombinanter Proteine durch *A. niger*. Dabei wird den Kultivierungsbedingungen, wie zum Beispiel

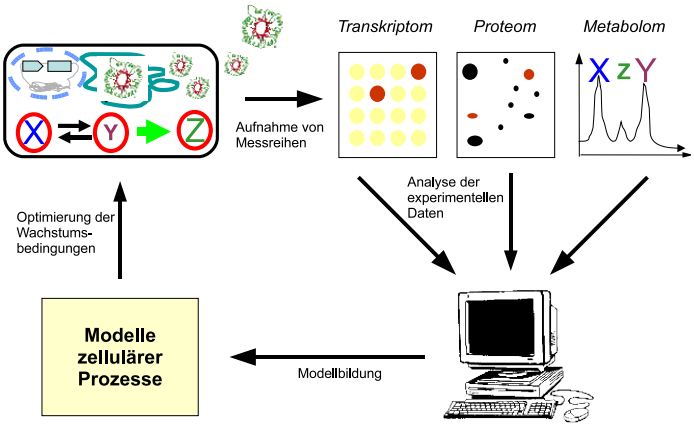

Abbildung 1.2: Dargestellt ist die Kombination aus Tanskriptom-, Proteom- und Metabolomdaten, die mittels bioinformatischer Methoden Vorhersagen über das biologische System erlaubt.

Temperatur, pH-Wert, Sauerstoffpartialdruck und Kohlenstoffquelle besondere Aufmerksamkeit geschenkt, da diese das Wachstum unmittelbar kontrollieren und damit einen erheblichen Einfluss auf die Proteinproduktion ausüben.

Neben der Klärung der Zusammenhänge zwischen Wachstumsbedingungen, Zellmorphologie und Produktivität wird im Teilprojekt B4 ein systembiologischer Ansatz verfolgt, der mit Hilfe der verschiedenen „Omics-Techniken“ die Parameter bestimmt, die für die Produktion entscheidend sind. Zu diesem Zweck werden Un-

tersuchungen des Transkriptoms, Proteoms und Metaboloms mit Techniken der Bioinformatik kombiniert, um die Daten effizient auszuwerten und schließlich Modelle zellulärer Teilprozesse zu generieren.

In einem iterativen Prozess soll das Modell letztendlich mit den experimentellen Daten angepasst und optimiert werden, um Aussagen über Produktionsengstellen treffen zu können, die dann aufgelöst werden können. In Abbildung 1.2 ist dieser Prozess schematisch dargestellt[1,2].

1.4 Die Schimmelpilzgattung *Aspergillus*

Die Gattung *Aspergillus* gehört zur Klasse der „Echten Schlauchpilze (Ascomycetes)“. Ihnen gemein ist die Ausbildung eines schlauchähnlichen Gebildes (Ascus), in dem die Ascusspore gebildet wird. Die Ascusspore stellt das Endprodukt der sexuellen Fortpflanzung der Ascomyceten dar. Die Ascomyceten zeichnen sich außerdem durch ein septiertes Myzel aus [66]. Die Gattung der Aspergilli umfasst sowohl Arten, die saprotroph leben, als auch opportunistische Krankheitserreger. Alle Krankheiten, an denen *Aspergillus*-Arten beteiligt sind, werden als Aspergillose bezeichnet. Im Allgemeinen versteht man darunter seltene Infektionen durch *Aspergillus*, die sich meist in der Lunge oder den Ohren manifestieren. Meist sind daher auch Symptome am Atmungsapparat festzustellen, jedoch können sich auch Läsionen an Leber, Darm oder anderen Stellen des Körpers bilden [69].

Aspergilli sind in der Natur weit verbreitet und kommen im häuslichen Bereich auf altem Brot, Käse oder Obst vor. Sie wachsen als filamentöse Pilze. Jedes Filament wächst hauptsächlich an der Spitze durch Verlängerung der terminalen Zelle. Die einzelnen Filamente bezeichnet man als Hyphe. Durch multiples Verzweigen der Hyphen und dadurch, dass sich auf diese Weise viele verschiedene Hyphen miteinander verweben, entsteht ein Teppich aus Pilzfäden, den man als Myzel bezeichnet [56]. Verschiedene *Aspergillus*-Arten finden in der Biotechnologie Verwendung, um zum Beispiel Zitronensäure, Sojasoße oder Essig herzustellen. Auch in der genetischen Forschung gehören verschiedene *Aspergillus*-Arten mittlerweile zu den

[1] `http://sfb578.tu-braunschweig.de/seiten/tp/tpb4.html`
[2] `http://www.tu-braunschweig.de/ibvt/forschung/projekte/sfb578-b4`

etablierten Modellorganismen [1].

1.4.1 *Aspergillus niger*

Aspergillus niger hat seinen Namen von der der Farbe seiner Konnidien, der asexuellen Vermehrungsform der Ascomyceten, die im Allgemeinen tiefschwarz sind. Unter Kupfermangel jedoch verfärben sich diese Sporen zuweilen auch schon mal gelb [69]. Der filamentöse Pilz findet in der Biotechnologie viele Anwendungen. So wird *A. niger* beispielsweise eingesetzt, um Zitronensäure für die Lebensmittelindustrie oder Pharmabranche herzustellen. Außerdem wird mit Hilfe von *A. niger* Gluconsäure produziert, die als Träger von Kalzium oder Natrium ebenfalls in der Lebensmittelindustrie Verwendung findet.

Das Genom von *A. niger* wird auf eine Größe zwischen 35,5 und 38,5 Megabasenpaare geschätzt, die sich auf 8 Chromosomen verteilen [6]. Der GC-Gehalt wird mit 52% angegeben [47].

A. niger ist ein Bodenbewohner und hat als saprotroph lebender Organismus Anteil am globalen Kreislauf des Kohlenstoffs. Als Modellorganismus dient er unter anderem zur Untersuchung der eukaryotischen Proteinsekretion, des Einflusses verschiedener Umweltfaktoren auf das Ausscheiden biomasseabbauender Enzyme, der entscheidenden molekularen Mechanismen zur Entwicklung von Fermentationsprozessen und Regulationsstrukturen der Pilzmorphologie [6].

1.4.2 Weitere *Aspergillus*-Arten

Die Gattung *Aspergillus* umfasst mehr als 185 Arten [27]. Da die einzelnen Arten zum Teil eine große Bedeutung in der Biotechnologie oder als Krankheitserreger haben, wurden inzwischen die Genome von mehreren Arten aufgeklärt. Einige dieser *Aspergillus*-Arten werden im Folgenden kurz vorgestellt:

Aspergillus nidulans

Aspergillus nidulans hat große Bedeutung als Modellorganismus in der Genetik erlangt. Der Grund dafür beruht unter anderem darauf, dass er im Gegensatz zu vielen anderen *Aspergillus*-Arten einen gut untersuchten sexuellen Fortpflanzungszyklus

besitzt. Das Genom von *A. nidulans* ist ca. 30 Megabasenpaare groß und besitzt ca. 10000 proteinkodierende Sequenzen auf 8 Chromosomen. Der GC-Gehalt beträgt 50% [27].

Aspergillus fumigatus

Aspergillus fumigatus kann Allergien auslösen, als opportunistischer Krankheitserreger wirken oder auch primärer Krankheitsauslöser sein. Der Pilz ist sehr weit verbreitet und man findet ihn beispielsweise häufig in Häusern und Wohnungen aber auch in Komposthaufen. Das Genom von *A. fumigatus* ist 29,4 Megabasenpaare groß und auf 8 Chromosomen verteilt. Der GC-Gehalt beträgt 49,9% [58].

Aspergillus oryzae

Aspergillus oryzae hat besonders in Japan große wirtschaftliche Bedeutung, da er zum Beispiel eingesetzt wird, um Reis zu Wein zu fermentieren (Sake) oder Sojasoße herzustellen. Durch seine Fähigkeit große Mengen Enzym, wie zum Beispiel Amylasen und Proteasen, zu sekretieren, spielt er auch bei der heterologen Proteinproduktion eine Rolle [46]. Das Genom von *A. oryzae* ist 37,2 Megabasenpaare groß und ebenfalls auf 8 Chromosomen verteilt. Der GC-Gehalt beträgt 48,2% [51].

1.5 Heterologe Proteinexpression

Unter heterologer Proteinexpression versteht man die Übertragung eines Fremdgens in einen Wirtsorganismus und die Expression des Gens, mit der Absicht, dass das entsprechende Protein von dem Wirtsorganismus synthetisiert wird. Diese Technik ist in der Biotechnologie weit verbreitet und bildet die Grundlage für die Herstellung vieler Stoffe für die Pharma- und Lebensmittelindustrie. So beruhen beispielsweise biotechnologische Verfahren zur Herstellung von Insulin, vieler Antibiotika, Ascorbinsäure, Chymosin (Enzym des Labferments), etc. auf dieser Technik.

Um einen Wirtsorganismus zur Produktion eines für ihn fremden Proteins zu bewegen, sind mehrere Schritte erforderlich. Zunächst muss das Zielgen aus dem Organismus isoliert werden. Hat man das Zielgen isoliert wird der Wirtsorganismus

mittels eines geeigneten Vektors mit dem Zielgen transformiert, so dass die Gensequenz nun zum Ablesen im Wirt vorliegt. Anschließend sollte der Wirt das gewünschte Protein synthetisieren und gegebenenfalls ins Medium sekretieren. Handelt es sich um einen Wirt, der das Produkt nicht ins Medium sekretiert, schließen sich noch ein Zellaufschluss und ein Reinigungsschritt an (siehe Abbildung 1.3).

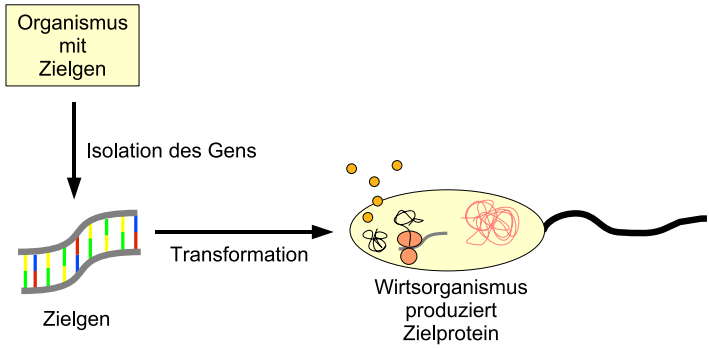

Abbildung 1.3: Schematische Darstellung der heterologen Proteinproduktion. Nachdem das Zielgen aus dem Organismus isoliert wurde, kann der Wirtsorganismus damit transformiert werden. Einige Wirtsorganismen sekretieren das gewünschte Zielprodukt direkt ins Medium. Andernfalls schließt sich ein Zellaufschluss und ein Reinigungsschritt an.

Bei dieser stark vereinfachten Beschreibung heterologer Proteinproduktion können eine Reihe von Problemen auftreten. Hat man erstmal das Gen isoliert und konnte die Sequenz stromabwärts eines geeigneten Promotors kloniert werden, wird der Wirtsorganismus mit dem Zielgen transformiert. Wenn dieser Schritt erfolgreich verläuft, ist das leider noch keine Garantie dafür, dass das Zielprodukt synthetisiert wird. Ein wichtiger Punkt der berücksichtigt werden muss ist, dass das Gen außerhalb seines natürlichen genomischen Kontextes steht. Auf die in Abbildung 1.3 dargestellte Vorgehensweise werden beispielsweise regulatorische Elemente, die sich stromaufwärts oder stromabwärts des betrachteten Gens befinden, gänzlich außer Acht gelassen. Weiterhin muss man berücksichtigen, dass der Wirtsorganismus und der Organismus, aus dem das Gen stammt, unter Umständen unterschiedliche regulatorische Elemente bevorzugen. Letztlich kann sich auch der verwendete geneti-

sche Code in einigen Punkten unterscheiden [31].

Auf eine besondere Art der Regulation innerhalb der Sequenz und die damit verbundenen Schwierigkeiten der heterologen Proteinexpression, wird im Folgenden eingegangen: die „codon usage“.

1.5.1 Kodonnutzung und heterologe Proteinproduktion

Der genetische Code ist degeneriert, das bedeutet, dass bis zu sechs Kodons für die gleiche Aminosäure kodieren können. Abhängig von dem jeweiligen Organismus, werden die Kodons unterschiedlich häufig für die betrachtete Aminosäure eingesetzt [36, 37]. Beispielsweise verwenden GC-reiche Organismen mehr Kodons die ‚G‘ und ‚C‘ enthalten als Kodons die ‚A‘ und ‚T‘ enthalten. Aus dieser Bevorzugung bestimmter Kodons ergibt sich, dass es optimale und suboptimale Kodons für jeden Organismus gibt. Die „codon usage“ verschiedener Gene eines Organismus hat großen Einfluss auf die Expressivität des Gens [30]. Die Bevorzugung bestimmter Kodons ist nicht universell gültig, sondern für jeden Organismus einzigartig. Der Grund für dieses Ungleichgewicht der einzelnen Kodons beruht auf der unterschiedlichen Anzahl der entsprechenden tRNAs in den verschiedenen Organismen.

Die „codon usage“ spielt somit eine entscheidende Rolle bei der heterologen Proteinexpression. Kodons im Zielgen, die in dem Wirtsorganismus selten eingesetzt werden, können zu einer geringen Translationsrate der mRNA, einer verringerten mRNA-Stabilität und manchmal so gar zu einem verfrühten Abbruch der Translation führen [72, 29]. In *Escherichia coli* wurde festgestellt, dass falsche Aminosäuren eingebaut werden können, wenn seltene Kodons für die Aminosäure Arginin verwendet werden [15].

Prinzipiell gibt es zwei mögliche Lösungen für dieses Problem. Die erste Möglichkeit ist es dem Wirtsorganismus die entsprechenden tRNAs zuzuführen, die andernfalls nur in unzureichenden Mengen vorliegen [14]. Die andere Möglichkeit, auf die hier näher eingegangen wird, besteht darin die Gensequenz „de novo“ zu synthetisieren und damit den Gegebenheiten des Wirtsorganismus anzupassen.

1.5.2 Kodonadaptierung

Kodonadaptierung meint die Anpassung einer fremden Gensequenz, im Folgenden „Zielgen“ genannt, an das Umfeld des Wirtsorganismus bezüglich der verwendeten Kodons. Dabei werden zuerst die optimalen Kodons für einen Wirtsorganismus be-

Tabelle 1.2: Bioinformatische Werkzeuge für die Analyse und/oder Anpassung der „codon usage“ an einen Wirtsorganismus.

Name	Referenz
Webserver	
DNAWorks	Hoover & Lubkowski [34]
GeneDesign	Richardson *et al.* [62]
IBG GeneDesigner	Vogelbacher *et al.* [75]
Optimizer	Puigbò *et al.* [61]
Synthetic Gene Designer (SGD)	Wu *et al.* [76]
Eigenständige Programme	
Codon Optimizer	Fuglsang [25]
GeMS	Jayaraj *et al.* [39]
GeneDesigner	Villalobos *et al.* [74]
UpGene	Gao *et al.* [28]

stimmt. Anschließend schreibt man das Zielgen so um, dass nur noch die optimalen Kodons des Wirtsorganismus in der Sequenz verwendet werden. Eine Variante der oben beschriebenen Anpassung ist, dass nicht alle, sondern nur ein Teil der Kodons ausgetauscht werden. Meist werden dann die Kodons ausgetauscht, die nach dem gewählten Kriterium besonders schlechte Werte haben. Für die Quantifizierung der „codon usage“ existieren verschiedene Indizes. Sie berechnen sich aus der Anzahl der verwendeten Kodons für die jeweilige Aminosäure im betrachteten Organismus. Als Beispiele seien hier der „Codon Adaptation Index (CAI)“ [68], die „Frequency of optimal codons (Fop)“ [48] und der „Codon Bias Index (CBI)“ [8] genannt. Als Zwischenschritt, bei der Berechnung der “codon usage“ für ein Gen, werden für die einzelnen Kodons Werte festgelegt, die dann im zweiten Schritt für alle Kodons

des betrachteten Gens aufsummiert oder multipliziert werden. Die Werte, die für die Kodons festgelegt wurden, werden oft als Maß für die Güte eines Kodons im jeweiligen Organismus verwendet. Auch die Anzahl der Kopien der tRNA im Genom für ein spezielles Kodon, wird manchmal als Maß für die Güte eines Kodons herangezogen [61].

Zur Analyse und Anpassung der „codon usage“ einer Sequenz an einen bestimmten Wirtsorganismus, stehen eine Reihe bioinformatischer Werkzeuge zur Verfügung. Einige dieser Werkzeuge sind in Tabelle 1.2 dargestellt. Die deutliche Zunahme von Werkzeugen in diesem Bereich, innerhalb der letzten zwei Jahre, ist damit zu erklären, dass die Preise für die Herstellung synthetischer Gene kontinuierlich fallen. Die Herstellung synthetischer Gene hat sich also zu einer echten Alternative zur Verwendung kodonoptimierter Wirtsorganismen entwickelt.

1.6 Aufgabenstellung

Eine wichtige Aufgabe des SFB 578 „Vom Gen zum Produkt“ ist die modellhafte Erschließung von *Aspergillus niger* als eukaryotischem Produzenten von Enzymen. Im Teilprojekt B4 „Systembiologie der Produkt- und Pelletbildung durch *Aspergillus niger*“ sollen hierfür die theoretischen und experimentellen Grundlagen geschaffen werden. Im Laufe der vorliegenden Arbeit wurden drei verschiedene *A. niger* Genomsequenzen zugänglich. Um die Ergebnisse dieser Genomdaten innerhalb des SFBs verfügbar zu machen, sollte ein netzwerkbasiertes System implementiert werden, das eine erweiterte Annotation, sowie das Ableiten metabolischer und regulatorischer Wege erlaubte. Das System sollte mit experimentellen Daten, insbesondere Transkriptom-, Proteom- und Metabolomdaten erweitert werden können und damit die Grundlage für den systembiologischen Ansatz des Projektes darstellen.

Neben der konkreten Problemstellung, *A. niger* als Wirtsorganismus zu etablieren, sollten auch allgemeine Probleme der heterologen Proteinexpression in dieser Arbeit bearbeitet werden. Ein wichtiger Punkt bei der heterologen Proteinexpression ist die Kodonnutzung. Etablierte Werkzeuge zeigten deutliche Nachteile in Bezug auf ihre Plattformunabhängigkeit, Anwendbarkeit auf vielfältige Wirtsorganismen oder ihre benutzerfreundliche Bedienung. Aus diesem Grund sollte ein neues Werkzeug entwickelt werden, mit dessen Hilfe es auf einfache Art möglich ist, die Kodonnutzung an verschiedene Organismen anzupassen, insbesondere an die innerhalb des SFB 578 im Fokus stehenden Organismen *A. niger* und *Bacillus megaterium*. Eine Überprüfung, der mit Hilfe des bioinformatischen Werkzeugs gemachten Prognosen, sollte in verschiedenen Gruppen des SFBs erfolgen.

2 Material und Methoden

2.1 Annotation von Genomen

Die Annotation von Genomen bezeichnet die Schritte, die nötig sind, um einer nackten Nukleotidsequenz ihre Gene und deren Funktion zuzuordnen. Die Programme, die für die vorliegende Arbeit verwendet wurden, werden im Folgenden vorgestellt.

2.1.1 Gene Locator and Interpolated Markov ModelER (GLIMMER)

Das Programm „GLIMMER“ wurde zunächst entwickelt, um Gene in bakteriellen Sequenzen vorherzusagen [22]. Später wurde das Programm dann weiterentwickelt, um auch Nukleotidsequenzen von Eukaryonten analysieren zu können [65, 52, 53]. In der vorliegenden Arbeit wurde das Programm in der Version „Glimmer-HMM“ dazu verwendet, „Open Reading Frames (ORF)“ aus unannotierten Nukleotidsequenzen vorherzusagen.

GlimmerHMM basiert auf einem „generalisiertem Hidden Markov Model“. Ein „Hidden Markov Model“ ist ein stochastisches Modell, das, nachdem es mit einem geeignetem Datensatz trainiert wurde, Vorhersagen erlaubt, in wie weit eine Eingabesequenz dem Trainingssatz entspricht. Die Eingabesequenz wird dabei als eine „Markov-Kette“ mit unbekannten Parametern verstanden. Die Markov-Kette ist eine Abfolge von verschiedenen Zuständen, deren Übergänge durch Wahrscheinlichkeitswerte charakterisiert sind. Für eine DNA-Sequenz kann das zum Beispiel bedeuten, dass die Zustände bestimmte Basen sind. Aufgrund der Reihenfolge in der diese Basen in der Sequenz auftreten wird dann bestimmt, ob es sich bei der eingegebenen Sequenz beispielsweise um ein Gen handelt. Das generalisierte Hidden Markov Model geht noch einen Schritt weiter. Statt nur einzelner Symbole werden hier, bei jedem Übergang von einem Zustand zum nächsten, funktionelle Einheiten der DNA oder Sequenzabschnitte, wie zum Beispiel ein Exon bewertet [53].

Das Programm wird mit zwei Trainingssätzen ausgeliefert. Der erste Trainingssatz besteht aus den annotierten Genen von *Arabidopsis thaliana*. Der zweite Trainingssatz, der für die vorliegende Arbeit verwendet wurde, besteht aus Genen von *Aspergillus spec.*

2.1.2 Basic Local Alignment Search Tool (BLAST)

„BLAST“ ist ein Programmpaket, das verschiedene Programme zum datenbankgestütztem Vergleich von Nukleotid- und Proteinsequenzen bereitstellt. Mit dem Programm „blastp“ wurden in der vorliegenden Arbeit die Aminosäuresequenzen, der mit GlimmerHMM vorhergesagten und translatierten, ORFs, mit Proteinsequenzen bekannter Funktion verglichen, um Rückschlüsse auf die Funktion des potentiellen Proteins ziehen zu können. BLAST verwendet zu diesem Zweck einen heuristischen Suchalgorithmus. Neben dem Ergebnis der Suche werden statistische Werte berechnet, die die Qualität der Treffer beschreiben.Für den Datenbankabgleich bietet BLAST eine Reihe von Programmen, die in Tabelle 2.1 dargestellt sind. Bei

Tabelle 2.1: Verschiedene BLAST-Programme (Quelle: `http://www.ncbi.nlm.nih.gov/BLAST/`).

Name	Funktion
Nukleotid-BLAST	Eine Nukleotiddatenbank wird nach einer Nukleotidsequenz durchsucht. Algorithmen: blastn, megablast, discontiguous megablast
Protein-BLAST	Eine Proteindatenbank wird nach einer Proteinsequenz durchsucht. Algorithmen: blastp, psi-blast, phi-blast
blastx	Eine Proteindatenbank wird nach einer vom Programm translatierten Nukleotidsequenz durchsucht.
tblastn	Eine vom Programm translatierte Nukleotiddatenbank wird nach einer Proteinsequenz durchsucht.
tblastx	Eine vom Programm translatierte Nukleotiddatenbank wird mit einer vom Programm translatierten Nukleotidsequenz durchsucht.

der BLAST-Suche wird zunächst eine Tabelle mit kurzen Teilsequenzen der Datenbanksequenzen als auch der Suchsequenz erstellt. Anschließend können mit diesen Tabellen die Teilsequenzen der Suchsequenz in der Datenbank sehr schnell gefunden werden. Diese initialen Treffer, die noch keine Lücken enthalten dürfen, werden

dann in mehreren Schritten zu beiden Seiten verlängert, bis das größtmögliche Alignment für diese beiden Sequenzen gefunden wird. Dieses Alignment darf auch Lücken enthalten. Der „BitScore“ ist ein für die Länge des Alignments normalisierter Wert für die Qualität des Alignments. Die statistische Signifikanz des Treffers wird durch den „E-Value“ angegeben [2, 3].

2.1.3 Metabolic Search and Reconstruction Kit (metaSHARK)

Das Programmpaket „metaSHARK“ dient der Vorhersage von enzymkodierenden Genen in nicht annotierten Nukleotidsequenzen. Zu diesem Zweck werden verschiedene Programme kombiniert, die nacheinander ausgeführt werden. Im ersten Schritt

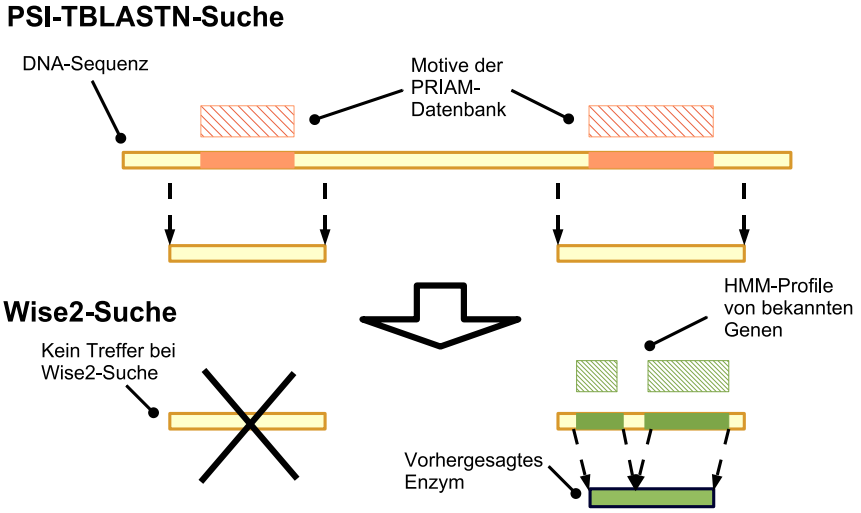

Abbildung 2.1: Dargestellt ist hier der Ablauf der Genomannotation mit dem Programmpaket „metaSHARK“ nach Pinney *et al.* [60]. Beschreibung siehe Text.

wird die DNA-Sequenz nach speziellen Motiven durchsucht, die aus der „PRIAM-Datenbank“ stammen. Die PRIAM-Datenbank enthält die wesentlichen Motive bekannter Enzyme als „positions-spezifische, gewichtete Matrix“ [20]. Die Suche erfolgt dabei mit Hilfe des Programms „PSI-TBLASTN“ [3]. Mit PSI-TBLASTN ist

eine sehr sensitive Suche von kodierenden Regionen in DNA-Sequenzen möglich. Diese Suche ist ein erster Filterungsschritt, um interessante Regionen in der DNA zu lokalisieren. Im nächsten Schritt werden die identifizierten Regionen extrahiert und mit dem Programm „Wise2“ analysiert. Wise2 vergleicht diese Sequenzen mit bereits bekannten Genen und sagt mit Hilfe eines Hidden Markov Models vorher, inwieweit diese Sequenzen den vorgegebenen ähneln. Bei diesem Schritt wird außerdem die Intron-Exon-Struktur des Gens bestimmt [13]. Anschließend wird die Proteinsequenz des potentiellen Proteins vorhergesagt. Diese Sequenz wird dann mit einer zweiten BLAST-Suche mit den Motiven aus der PRIAM-Datenbank verglichen. Die zweite BLAST-Suche liefert auf diese Weise auch statistische Werte für den Treffer [60]. In Abbildung 2.1 ist der Ablauf der Vorhersage schematisch dargestellt.

2.1.4 tRNAscan-SE

„tRNAscan-SE“ ist ein Programm zur Vorhersage von tRNA-Genen in einer Nukleotidsequenz. Das Programm ist im Internet frei verfügbar[1] und lässt sich auf jedem Linux-System installieren. Als Eingabe erwartet das Programm Nukleotidsequenzen im „FASTA-Format“. Das FASTA-Format besteht aus zwei Typen von Zeilen:

1. Bezeichnerzeile - Die Zeile beginnt mit dem Zeichen ‚>‘ gefolgt von der Bezeichnung der Sequenz. Diese Zeile wird allgemein „Header“ genannt.

2. Sequenzzeile - In dieser Zeile steht die Sequenz, die zum Beispiel eine Nukleotid- oder Aminosäuresequenz sein kann.

Auf die in dieser Form eingegebenen Sequenzen werden nacheinander drei verschiedene Algorithmen zur Vorhersage von tRNA-Genen angewendet. tRNAscan-SE findet mit dieser Methode 99-100% korrekter tRNA-Gene [50].

In der vorliegenden Arbeit wurde das Programm tRNAscan-SE dazu verwendet um tRNA-Gene aus den Nukleotidsequenzen von *Aspergillus niger* vorherzusagen.

[1] `ftp://ftp.genetics.wustl.edu/pub/eddy/software/tRNAscan-SE.tar.Z`

2.2 Datenquellen der *Aspergillus*-Datenbank

Für die *Aspergillus*-Datenbank wurden Daten aus verschiedenen Quellen bearbeitet und in die Datenbank integriert. Im Folgenden werden die Datenquellen vorgestellt.

2.2.1 Quellen der Genomdaten verschiedener *Aspergillus*-Arten

Die Sequenzen, für die in der *Aspergillus*-Datenbank gespeicherten Genome, stammen aus folgenden Quellen:

2.2.1.1 Integrated Genomics

Integrated Genomics Inc. ist eine amerikanische Firma mit Sitz in Chicago, Illinois. Im Rahmen des SFB 578 wurden durch das Teilprojekt Z - „Zentrale Aufgaben für den Sonderforschungsbereich“ - Nutzungsrechte an einer *Aspergillus niger*-Genomsequenz von Integrated Genomics erworben.

Bei der erworbenen Sequenz handelt es sich um den Stamm NRRL3. Neben der Bezeichnung NRRL3 wird der Stamm auch mit DSM 2466 oder ATCC 9029 bezeichnet. Der Stamm wird unter anderem zur Herstellung von Enzymen und organischen Säuren, beispielsweise Zitronensäure, verwendet. Die erworbene Sequenz besteht aus 33,69 Megabasen DNA-Sequenz, die auf 9510 „Contigs “ verteilt sind. „Contigs“ bezeichnen überlappende DNA-Fragmente aus verschiedenen sequenzierten Teilstücken („Reads“), die zu größeren Einheiten zusammengefasst wurden. Als Maß für die Qualität der Sequenzierung von Genomen kann man das „Coverage“ nach der der folgenden Formel berechnen:

$$Coverage = \frac{NL}{G} \tag{2.1}$$

N = Anzahl der sequenzierten Teilstücke („Reads“)
L = durchschnittliche Länge der sequenzierten Teilstücke
G = Länge des Genoms

Für die vorliegende Sequenz errechnet sich ein dreifaches Coverage. Aufgrund der Beschaffenheit des Genoms werden bei einem Coverage von drei nicht alle Bereiche des Genoms getroffen. So lassen sich beispielsweise repetitive Bereiche im

Genom nur schwer sequenzieren. Grundsätzlich gilt, je höher das Coverage, umso genauer ist das Ergebnis der Sequenzierung.

2.2.1.2 Joint Genome Institute (JGI)

Das Joint Genome Institute ist ein Zusammenschluss von fünf amerikanischen Instituten mit dem Ziel verschiedene Organismen, mit Hilfe von Sequenzierungsprojekten und computergestützten Analysen zu untersuchen, die für Energie und Umwelt eine Rolle spielen[2]. Ergebnisse von Sequenzierungsprojekten werden, auch wenn sie noch einen vorläufigen Charakter haben, der wissenschaftlichen Welt sofort zur Verfügung gestellt.

Im Jahr 2006 wurde vom JGI die vorläufige Sequenz von *A. niger* ATCC 1015 veröffentlicht. Die Sequenz besteht aus 37,19 Megabasen DNA-Sequenz, die auf 143 Contigs verteilt sind. Das Coverage ist 8,9-fach für diese Sequenz.

2.2.1.3 DSM

Die DSM ist eine niederländische Firma mit Sitz in Heerlen. Ihr Betätigungsfeld liegt in der Biowissenschaft und der Materialkunde. Die DSM beliefert sowohl die Lebensmittelbranche und die Pharmabranche mit Stoffen, die zum Beispiel in Fermentationsprozessen gewonnen werden können. Mit Produkten aus der Sparte der Materialkunde werden Kunden in der Automobilindustrie, Kunden von Sport- und Freizeitbranche, etc. beliefert[3].

Der Vorfahre eines für die Herstellung von Enzymen verwendeten Stammes von *A. niger* wurde im November 2006 von der DSM veröffentlicht. Die Sequenz des Stamms mit der Bezeichnung CBS 513.88 besteht aus 33,9 Megabasen DNA-Sequenz, die auf 468 Contigs verteilt sind. Das Coverage für diese Sequenz ist 7,5-fach.

[2] `http://www.jgi.doe.gov`
[3] `http://www.dsm.com`

2.2.1.4 Broad Institute

Das Broad-Institut ist ein Zusammenschluss des „Massachusetts Institute of Technology (MIT)“, Harvard und angeschlossener Krankenhäuser und des Whitehead-Instituts. Ziel ist es eine Brücke zwischen Medizin und Methoden der Genomanalyse zu schlagen. Am Broad-Institut werden verschiedene Projekte zu vergleichenden Genomanalysen durchgeführt. Zu diesem Zweck werden Daten aus Genomprojekten gesammelt und analytisch aufbereitet. Zum Teil werden aber auch eigene Sequenzierungsprojekte durchgeführt[4].

Für vergleichende Analysen an *Aspergillus* wurde eine Datenbank ins Netz gestellt, die unter anderem die reinen Sequenzdaten verschiedener *Aspergillus*-Arten zum Herunterladen anbietet. Aus dieser Quelle wurden die Sequenzdaten von *A. fumigatus*, *A. nidulans* und *A. oryzae* übernommen. Details zu den einzelnen Genomsequenzen sind in Tabelle 2.2 angegeben.

Tabelle 2.2: *Aspergillus*-Sequenzen aus der Datenbank des Broad-Instituts (`http://www.broad.mit.edu/annotation/genome/aspergillus_group`).

Name	Grösse in Megabasen	Coverage	Anzahl der Contigs bzw. Chromosomen
Aspergillus fumigatus	29,4	nicht angegeben	8
Aspergillus nidulans	30	13-fach	248
Aspergillus oryzae	37,2	9-fach	89

2.2.2 Weitere Quellen für die Erstellung der *Aspergillus*-Datenbank

Neben den beschriebenen Quellen, aus denen vor allem die Genomdaten stammen, wurden Daten aus weiteren Quellen in die Datenbank integriert, die im Folgenden beschrieben werden.

[4] `http://www.broad.mit.edu/`

2.2.2.1 Die „Kyoto Encyclopedia of Genes and Genomes (KEGG)“

Der Schwerpunkt von KEGG liegt auf der Analyse genomischer Daten. Ziel ist es mittels geeigneter bioinformatischer Methoden Vorhersagen für das Verhalten biologischer Systeme höherer Ordnung, wie zum Beispiel der Zelle oder des Organismus, aus den Genomdaten treffen zu können[5].

Das System von KEGG besteht aus mehreren Bereichen nämlich „Pathway“, „Genes“, „Ligand“ und „BRITE“. Die Abteilung „Pathway“ enthält Informationen zu metabolischen Pfaden und auch gezeichnete Karten dieser biologischen Prozesse. Der „Genes “-Bereich enthält genomische Daten. In „Ligand“ sind die Informationen zu chemischen Reaktionen und Enzymen zusammengefasst. Eine Sammlung hierarchischer Regeln für die Darstellung biologischer Systeme ist in der Abteilung „BRITE“ gespeichert [41].

Für die Erstellung der *Aspergillus*-Datenbank wurden vor allem die Daten aus der Ligand-Datenbank verwendet, um das metabolische Netzwerk der einzelnen Stämme zu konstruieren. Für die Darstellung von Stoffwechselwegen auf der Webseite werden außerdem Daten zur Laufzeit aus der KEGG-Datenbank abgefragt und verwendet (siehe 2.5.7, Seite 35).

2.2.2.2 Swiss-Prot und TrEMBL

Swiss-Prot und TrEMBL bilden zusammen eine mächtige Proteindatenbank. Es handelt sich dabei um ein Konglomerat von experimentellen und vorhergesagten Proteinen. Während die Swiss-Prot-Datenbank manuell annotiert wird und viele zusätzliche Informationen bietet, ist die TrEMBL eine Ergänzung der Swiss-Prot, die die Translation aller, in der EMBL-Datenbank vorhandenen, Nukleotidsequenzen enthält. Die Swiss-Prot bietet neben der reinen Sequenz noch Informationen bezüglich der Funktion des Proteins, Informationen zu posttranslationalen Modifikationen, wie zum Beispiel Acetylierung oder Phosphorylierung, Informationen über Domänen oder Bindestellen, Sekundärstrukturen, Ähnlichkeiten zu anderen Proteinen, etc. Zum Zeitpunkt der Erstellung dieser Arbeit werden in der Swiss-Prot-Datenbank Informationen zu 285335 Proteinen gespeichert. Die TrEMBL-Daten-

[5] `http://www.genome.jp/kegg/`

bank enthält zu diesem Zeitpunkt die Sequenzen von 4932421 Proteinen[6] [5].

Die Informationen der Swiss-Prot/TrEMBL-Datenbanken wurden in der vorliegenden Arbeit dazu verwendet, die mit GLIMMER (siehe 2.1.1, Seite 15) vorhergesagten potentiellen Proteine mittels BLAST (siehe 2.1.2, Seite 16) einer Funktion zuordnen zu können.

2.3 Optimierung heterologer Proteinexpression durch Kodonadaptierung

Um die „codon usage“ eines Gens an eine neue Umgebung anzupassen, sind eine Reihe von Berechnungen nötig. Es gibt verschiedene Möglichkeiten, um festzustellen wie gut die verwendeten Kodons an die Umgebung angepasst sind. In diesem Abschnitt werden die Algorithmen beschrieben, die das Programm „JCat“ verwendet, um diese Berechnungen durchzuführen.

2.3.1 Codon Adaptation Index (CAI)

Der „Codon Adaptation Index“ beschreibt wie ähnlich die Zusammensetzung der Kodons eines Gens der Zusammensetzung der Kodons eines Organismus ist. Für die Berechnung des Index muss eine Referenzmenge hochexpremierter Gene des betrachteten Organismus definiert werden. Mit Hilfe dieser Liste wird mit der in 2.2 angegebenen Formel eine Referenztabelle des „Relative Synonymous Codon Usage ($RSCU$)“ gebildet.

$$RSCU_{ij} = \frac{x_{ij}}{\frac{1}{n_i}\sum_{j=1}^{n_i} x_{ij}} \tag{2.2}$$

x_{ij} bezeichnet die Häufigkeit des Auftretens von Kodon j für die Aminosäure i. n_i ist die Anzahl der alternativen Kodons für die Aminosäure i. Gültige Werte für die Anzahl der alternativen Kodons sind Zahlen von eins bis sechs, die vom verwendeten genetischem Code des Organismus abhängen.

[6] http://expasy.org/sprot/

Aus dem $RSCU$ berechnet man im zweiten Schritt nach Formel 2.3 die „Relative Adaptiveness (w_{ij})“ eines Kodons.

$$w_{ij} = \frac{RSCU_{ij}}{RSCU_{imax}} = \frac{x_{ij}}{x_{imax}} \tag{2.3}$$

$RSCU_{imax}$ und x_{imax} bezeichnen den jeweiligen $RSCU$- bzw. x-Wert für das Kodon, das für die Aminosäure i im Referenzdatensatz am häufigsten verwendet wird. Die „Relative Adaptiveness (w_{ij})“ wurde im Programm „JCat“ als Maß für die Qualität eines Kodons verwendet.

Aus w_{ij} kann man anschließend den CAI nach Gleichung 2.4 für ein Gen berechnen.

$$CAI = (\prod_{k=1}^{L} w_k)^{\frac{1}{L}} \tag{2.4}$$

Der CAI dient als theoretisches Maß für die Expressivität eines Gens und kann Hinweise darauf liefern ob ein Gen in einem fremden Kontext exprimiert werden kann [67].

Neben dem hier beschriebenen Codon Adaptation Index sind auch noch andere Indizes gebräuchlich, um die „codon usage“ zu quantifizieren (siehe 1.5.1, Seite 10). Der CAI ist jedoch wohl der am weitesten verbreitete Index zur theoretischen Berechnung der Expressivität eines Gens [25].

2.3.2 Algorithmus zur iterativen Berechnung des CAI

Im Jahr 2003 wurde ein Algorithmus veröffentlicht, der aus einer Menge von Genen diejenigen Gene herausfindet, deren „codon usage“ die der restlichen Gene dominiert. Die Berechnung erfolgt dabei ohne menschliche Interaktion und die gefundenen Gene können als Referenzmenge für die Berechnung von CAIs genutzt werden [16].

Um diese Gene für einen Organismus zu berechnen geht man folgendermaßen vor: Im ersten Schritt wird das „Relative Synonymous Codon Usage ($RSCU$)“ berechnet. Als Referenzdatensatz dienen dabei alle Gene des betrachteten Organismus. Aus dem $RSCU$ wird dann die „Relative Adaptiveness (w_{ij})“ berechnet. Anschließend wird für alle Gene der CAI berechnet (siehe 2.3.1, Seite 23). Im folgenden

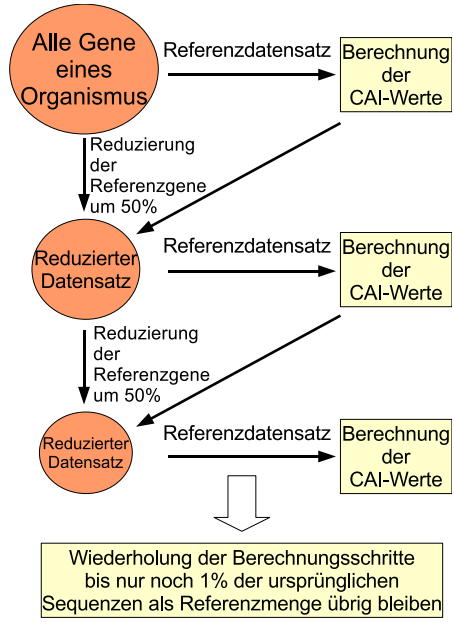

Abbildung 2.2: Dargestellt ist die Funktionsweise des Algorithmus zur Berechnung der dominierenden „codon usage“ nach Carbone *et al.* [16]. Nach jeder Berechnung der CAIs erfolgt eine Reduzierung der Referenzdatensatzes um 50%. Dabei werden nur die Gensequenzen übernommen, deren „codon usage“ dominierend ist. Der Algorithmus endet, wenn der Referenzdatensatz nur noch 1% der ursprünglichen Gensequenzen beinhaltet.

Schritt werden die 50% der Gene aus der Referenzmenge ausgewählt, die die dominierendste „codon usage“ aufweisen, d.h. die Gene mit den höchsten CAI. Diese Gene dienen nun im nächsten Durchlauf der Berechnung als Referenzmenge. Die Schritte werden so lange wiederholt, bis nur noch 1% von der Gesamtmenge der Gensequenzen des Organismus als Referenzmenge übrig bleibt (siehe Abbildung 2.2). Dass nur ca. 1% der ursprünglichen Sequenzen als Menge der Gene mit dominierendem „codon usage“ definiert wird, erfolgt in Anlehnung an die ursprüngliche Definition des CAI. Dort sind auch nur ca. 1% der Gensequenzen von *Escherichia coli* als Referenzdatensatz verwendet worden [67, 68].

Der Algorithmus wurde in der vorliegenden Arbeit dazu verwendet, um die Refe-

renzmenge hochexpremierter Gene für die Organismen vorherzusagen, die mit dem Programm „JCat“ optimiert werden können.

2.3.3 Algorithmus zur Vorhersage von rho-unabhängigen Transkriptions-Terminatoren

Bakterielle Genome sind unterteilt in Bereiche, die expremiert werden und Bereiche, die nicht expremiert werden. Diese Bereiche sind flankiert von Sequenzen, an denen die Transkription der DNA in RNA initiiert und terminiert wird. Die Genexpression kann unter anderem dadurch beeinflusst werden, dass die Effizienz des Initiationsprozesses und des Terminationsprozesses verändert wird. Ein Mittel dieser Art der Genregulation stellen rho-unbhängige Transkriptions-Terminatoren dar. Diese Sequenzen bilden eine „Haarnadelstruktur“ in RNA-Sequenzen aus und bewirken so einen Abbruch des Transkriptions-Prozesses (siehe Abbildung 2.3). Zur

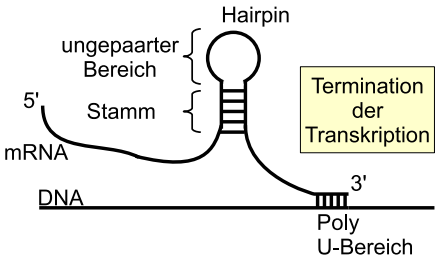

Abbildung 2.3: Dargestellt ist das Modell eines rho-unabhängigen Terminators nach Ermolaeva *et al.* [24].

Vorhersage dieser Sequenzen müssen verschiedene Dinge von der Sequenz bekannt sein:

1. Die Stabilität der RNA-Haarnadelstruktur
2. Zusammensetzung und Entfernung zur Uracil-reichen Region stromabwärts der Haarnadelstruktur
3. Position und Orientierung des Terminators in Bezug auf benachbarte Gene

Da in dieser Arbeit der Algorithmus lediglich zur Vermeidung unvorteilhafter Strukturen in der kodonoptimierten Sequenz verwendet wurde, wird auf den letztgenannten Punkt nicht näher eingegangen.

Zur Berechnung der Stabilität der RNA-Haarnadelstruktur wird folgende Formel verwendet:

$$E = gc \cdot x_1 + au \cdot x_2 + gu \cdot x_3 + mm \cdot x_4 + gp \cdot x_5 + lp \cdot x_6 - 5,7 \qquad (2.5)$$

x_1, x_2 und x_3 bezeichnen die Anzahl der Paarungen von G-C-, A-U- und G-U-Nukleotiden im Stamm der Haarnadelstruktur. x_4 und x_5 geben die Anzahl der Fehlpaarungen („mismatches“) und Lücken („gaps“) im Stamm an. Die Anzahl der Nukleotide im ungepaarten Bereich fließt als Variable x_6 in die Berechnung ein.

Für die Nukleotidpaarungen, Fehlpaarungen, Lücken und die Anzahl der Nukleotide im ungepaarten Bereich werden Werte eingefügt. Die besten Ergebnisse werden für die Folgenden erhalten [24]:

$$E = 2,3 \cdot x_1 + 0,9 \cdot x_2 + 1,3 \cdot x_3 + 3,5 \cdot x_4 + 6,0 \cdot x_5 + 1,0 \cdot x_6 - 5,7 \qquad (2.6)$$

Zur Bewertung des Uracil-reichen Bereichs stromabwärts der Haarnadelstruktur, wurde die von d'Aubenton Carafa *et al.* [21] vorgeschlagene Formel verwendet:

$$T = -\sum_{n=1}^{15} x_n \qquad (2.7)$$

Dabei gilt:

$x_n = x_{n-1} \cdot 0,9$, wenn das n-te Nukleotid ein Uracil ist.

$x_n = x_{n-1} \cdot 0,6$, wenn das n-te Nukleotid **kein** Uracil ist.

Für die vorliegende Arbeit wurde der „Cutoff-Value“, also die Schwelle an der die Bedingung für einen Haarnadelstruktur noch erfüllt ist, mit ‚-6,0‘ definiert. Der Cutoff-Value für den Uracil-reichen Bereich wurde mit ‚-2,0‘ festgelegt.

2.4 Datenquellen für die Kodonanpassung

Um heterologe Proteine zu exprimieren und die „codon usage“ des entsprechenden Gens anpassen zu können, muss das Genom des Wirtsorganismus analysiert werden, in dem das Protein synthetisiert werden soll. Zu diesem Zweck werden Daten

aus verschiedenen Datenquellen verwendet, die in diesem Abschnitt beschrieben werden.

2.4.1 Die Genome-Reviews-Datenbank

Die „Genome-Reviews-Datenbank“ hat es sich zum Ziel gemacht, die Darstellung kompletter Genomsequenzen zu standardisieren. Die große Zahl neu publizierter Genome aus unterschiedlichen Quellen macht es häufig schwierig für Entwickler eigene Datenbanken auf dem aktuellen Stand zu halten, da bereits kleine Fehler in einer Datei die Integration neuer Daten sehr komplex gestalten können. Auch inhaltliche Fehler in Dateien kommen gelegentlich vor. Ein weiteres Problem ist, dass die Integration experimentell erhaltener Daten bei der Veröffentlichung eines neuen Genoms oft unberücksichtigt bleibt [44, 71]. Aus diesem Grund integriert die Genome-Reviews-Datenbank Informationen aus verschiedenen Datenquellen, wie zum Beispiel EMBL und Swiss-Prot. Auch die Annotation der Daten wird gegebenenfalls nachbearbeitet. So werden zum Beispiel Gene die für tRNAs kodieren vorhergesagt, falls das bei der Publikation des neuen Genoms von den jeweiligen Autoren noch nicht durchgeführt wurde. Der Schwerpunkt der Datenbank liegt auf den Genomdaten von Prokaryonten, aber es sind auch einige ausgesuchte eukaryontische Sequenzen vorhanden, zum Beispiel *Saccharomyces cerevisae* und *Arabidopsis thaliana*.

Die Datenbank GenomeReviews wird zum Herunterladen in einfachen Textdateien angeboten[7]. Diese Dateien wurden in der vorliegenden Arbeit verwendet, um die „codon usage“ für Prokaryonten im Programm „JCat“ zu berechnen.

2.4.2 Weitere Datenquellen

Neben den bereits erwähnten Genome-Reviews wurden noch weitere Datenquellen für die Berechnung der „codon usage“ verwendet:

[7] `http://www.ebi.ac.uk/GenomeReviews`

Bacillus megaterium

Innerhalb des SFB 578 wurde die Sequenzierung von *Bacillus megaterium* DSM 319 durch die Firma GATC-Biotech AG durchgeführt. Die Annotation wurde 2005 im Rahmen einer Diplomarbeit innerhalb der Arbeitsgruppe Jahn begonnen [35]. Die Annotation ist noch nicht abgeschlossen und das Genom besteht zur Zeit aus 250 Contigs.

Arabidopsis thaliana

Die Genomsequenz von *Arabidopsis thaliana* stammt vom „The Institute for Genomic Research (TIGR)“[8]. Die Sequenz wurde anschließend vom „Munich Information Center for protein sequence (mips)“ annotiert und mit weiteren Daten ergänzt[9].

Aspergillus niger ATCC 1015

Die Genomsequenz zur Berechnung der „codon usage“ des Stamms *Aspergillus niger* ATCC 1015 stammt aus der gleichen Quelle wie in 2.2.1.2, Seite 20 angegeben.

Aspergillus niger NRRL3

Die Genomsequenz zur Berechnung der „codon usage“ des Stamms *Aspergillus niger* NRRL3 stammt aus der gleichen Quelle wie in 2.2.1.1, Seite 19 angegeben.

Caenorhabditis elegans

Die Genomsequenz zur Berechnung der „codon usage“ von *Caenorhabditis elegans* stammt aus der Datenbank „Wormbase“[10] [12].

Drosophila melanogaster

Die Genomsequenz zur Berechnung der „codon usage“ von *Drosophila melanogaster* stammt aus der Datenbank „Berkeley Drosophila Genome Project“[11] [18].

[8] `http://www.tigr.org/tdb/e2k1/ath1/`
[9] `http://mips.gsf.de/proj/plant/jsf/athal/index.jsp`
[10] `http://wormbase.org`
[11] `http://www.fruitfly.org`

Homo sapiens

Die Genomsequenz zur Berechnung der „codon usage“ für das humane Genom stammt vom „The German cDNA Consortium“[12] [38].

Mus musculus

Die Genomsequenz zur Berechnung der „codon usage“ für das Genom von *Mus musculus* stammt vom „NIA Mouse cDNA Project“[13] [17].

Saccharomyces cerevisae

Die Genomsequenz zur Berechnung der „codon usage“ von *Saccharomyces cerevisae* stammt aus der „Yeastgenome“-Datenbank[14] [19].

2.4.3 Translationstabellen vom „National Center of Biotechnology Information (NCBI)“

Das „National Center of Biotechnology Information (NCBI)“ bietet Translationstabellen für verschiedene Organismen bzw. Zellorganelle an. Mit den Tabellen ist es möglich Nukleotidsequenzen aus z.B. pflanzlichen oder bakteriellen Plastiden, Wirbeltiermitochondrien, etc. in eine Aminosäuresequenz zu übersetzen. Insgesamt stehen 17 Translationstabellen zur Verfügung[15].

Die Translationstabellen wurden in das Programm „JCat“ integriert, um Sequenzen einlesen zu können, die nicht dem Standardcode entsprechen.

2.4.4 The Restriction Enzyme Database (REBASE)

Die REBASE-Datenbank enthält umfassende Informationen zu Restriktionsenzymen, DNA-Methyltransferasen und verwandten Proteinen. Es sind sowohl publizierte als auch unpublizierte Referenzen, Erkennungssequenzen und Schnittstellen,

[12] http://mips.gsf.de/projects/cdna/german_human_project_index_html
[13] http://lgsun.grc.nia.nih.gov/cDNA/
[14] http://www.yeastgenome.org
[15] http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi

Informationen zur kommerziellen Verfügbarkeit, Kristallisations- und Sequenzdaten in der Datenbank gespeichert. Damit bietet REBASE die größte Datensammlung mit diesem Schwerpunkt [64, 63].

Anfang 2007 enthält die Datenbank nahezu 3800 Einträge. Die Datenbank ist als Textdatei verfügbar[16] und wurde als solche dazu verwendet, unerwünschte Restriktionsenzymbindestellen aus kodonoptimierten Sequenzen zu entfernen.

2.5 Verwendete Techniken aus der Informatik

In diesem Abschnitt werden Techniken der Informatik beschrieben, die entweder für die Erstellung der *Aspergillus*-Datenbank oder für das Programm „JCat“ verwendet wurden.

2.5.1 Das Datenbankmanagementsystem PostgreSQL

Datenbanken dienen zur Verwaltung großer Datenbestände. Mit ihrer Hilfe ist es möglich komfortabel auf Daten zuzugreifen, Daten zu verändern, neue Daten hinzuzufügen oder Daten zu löschen. Zugriff auf die Daten erfolgt dabei immer mit Hilfe eines Datenbankmanagementsystem, das die Datenorganisation übernimmt. In der vorliegenden Arbeit wurde PostgreSQL als Datenbanksystem für die *Aspergillus*-Datenbank gewählt. PostgreSQL ist ein relationales Datenbankmanagementsytem, das die Daten in Tabellen organisiert. In Abbildung 2.4 ist als Beispiel für eine Tabelle eines relationalen Datenbankmanagementystems die Tabelle ‚contig‘ der *Aspergillus*-Datenbank dargestellt. Diese Tabelle enthält die Nukleotidsequenzen der Contigs. Neben den Spaltennamen sind die jeweiligen Datentypen angegeben. Verbindungen zwischen den Tabellen bezeichnet man als Referenzen. Die Gesamtheit aller Tabellen und Referenzen nennt man Datenbankschema. Als Schnittstelle zwischen Benutzer und Datenbankmanagementsystem dient die „Structured Query Language (SQL)“. In Abbildung 2.5 ist dieser Zusammenhang schematisch dargestellt. **S**QL ist eine Abfragesprache für relationale Datenbanksysteme. Mit ihrer Hilfe können Daten sowohl abgefragt als auch manipuliert werden. Der von Post-

[16] `http://rebase.neb.com/rebase/rebase.ftp.html`

Contig	
Contig_No	*Integer*
Contig_Entry	*Varchar(15)*
Sequence	*Text*
Size	*Integer*
G_content	*Integer*
C_content	*Integer*
A_content	*Integer*
T_content	*Integer*
Comment	*Varchar(1000)*

Abbildung 2.4: Dargestellt ist hier die Tabelle eines relationalen Datenbankmanagementsystems. In der linken Spalte sind die Spaltennamen angegeben. Die rechte Spalte gibt den jeweiligen Datentyp an.

greSQL verwendete Dialekt von SQL hält sich weitgehend an die vom „American National Standards Institute“ festgelegten Standards „ANSI-SQL 92/99“[17].

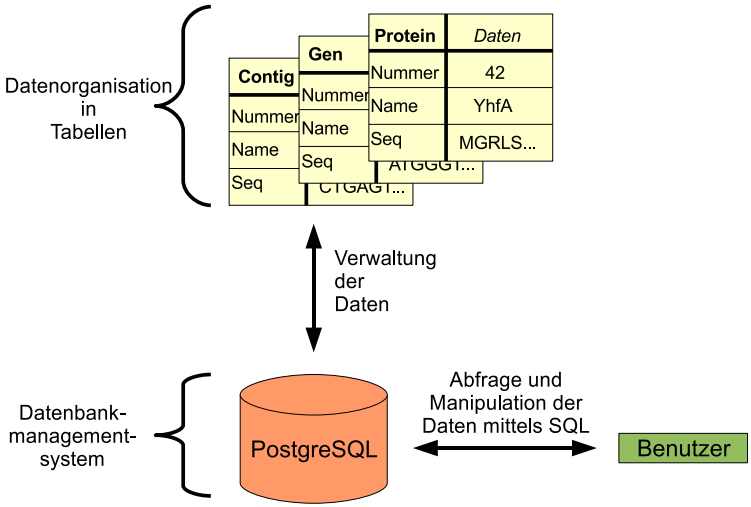

Abbildung 2.5: Dargestellt ist der Zusammenhang zwischen Datenorganisation, Datenbankmanagementsystem und Benutzerinteraktion bei der Verwendung eines relationalen Datenbankmanagementystems.

[17] `http://www.postgresql.org/`

2.5.2 Java

Java ist eine objektorientierte Programmiersprache. Sie ist plattformunabhängig, so dass Programme, die in dieser Sprache geschrieben wurden, ohne großen Aufwand auf allen Computern laufen, die über eine entsprechende Java-Laufzeitumgebung („Java-Runtime-Environment“) verfügen. Entwickelt wurde die Sprache Mitte der 90er von der Firma „Sun Microsystems“[18]. Seither hat die Sprache gerade in Koevolution mit dem Internet einen regelrechten „Boom“ erfahren. Mittlerweile ist Java die am weitesten verbreitete Programmiersprache, wenn man die Anzahl der in Java geschriebenen „Open-Source“-Projekte als Grundlage verwendet[19].

Die Programmiersprache ist stark strukturiert, so sind Klassen beispielsweise in Paketen angeordnet. Diese Pakete können in neue Projekte integriert werden und ermöglichen so eine unkomplizierte Einbindung bereits implementierter Funktionen in die neue Anwendung. Java ist aufgrund seiner umfangreichen Klassenbibliothek geeignet, um umfangreiche Programme mit graphischer Oberfläche, Webapplikationen, datenbankgestützte Anwendungen und viele weitere Programme zu entwickeln[20].

In der vorliegenden Arbeit wurde Java für alle implementierten Anwendungen als Programmiersprache eingesetzt.

2.5.3 Das „Java-Package“ JFreeChart

„JFreeChart“ ist eine Java-Programm-Bibliothek, mit deren Hilfe es einfach möglich ist Diagramme zu erstellen, die in jede Applikation eingefügt werden können. Die Bibliothek ist frei verfügbar, darf allerdings nicht verändert werden (GNU Lesser General Public License[21]). JFreeCharts herausragende Eigenschaften umfassen:

- eine gut dokumentierte Programmierschnittstelle (API), die viele verschiedene Diagrammtypen bietet
- ein flexibles Design, das einfach erweitert werden kann und sowohl für „serverseitige“ als auch „clientseitige“ Applikationen entwickelt wurde

[18] `http://www.sun.com/`
[19] `http://www.cs.berkeley.edu/~flab/languages.html`
[20] `http://java.sun.com/`
[21] `http://www.gnu.org/licenses/lgpl.html`

- Unterstützung für unterschiedliche Ausgabedateitypen zum Beispiel „Swing components“, verschiedene pixelbasierte Grafiken und verschiedene vektorbasierte Grafiken
- JFreeChart darf frei und ohne Beschränkungen verwendet werden[22]

In der vorliegenden Arbeit wurde JFreeChart dazu verwendet, um im Programm „JCat“ die „codon usage“ über der eingegebenen bzw. optimierten Sequenz darzustellen. Außerdem werden mit Hilfe von JFreeChart experimentelle Metabolomdaten, die in der *Aspergillus*-Datenbank gespeichert sind, dargestellt.

2.5.4 Das Java-Datenbankmanagementsystem HSQLDB

„HSQLDB“ ist ein relationales Datenbankmanagementsystem, das komplett in der Programmiersprache Java geschrieben ist. Es ist die Fortsetzung eines Projektes das als „HypersonicSQL“ begonnen wurde und hat daher auch seinen Namen erhalten.

HSQLDB bietet einen JDBC-Treiber, mit dessen Hilfe aus Java-Programmen heraus auf die Datenbank zugegriffen werden kann. Die Spezifikation und Syntax der Datenbank hält sich dabei weitgehend an die offiziell festgelegten Standards „ANSI-92“ und an Erweiterungen, die unter den Namen „SQL99“ und „SQL2003“ beschlossen wurden. Weitere Eigenschaften von HSQLDB sind:

- HSQLDB ist sehr klein und sehr schnell
- Tabellen einer Datenbank können auf der Festplatte gespeichert werden oder nur im Hauptspeicher gehalten werden
- HSQLDB enthält verschiedene Werkzeuge zum Zugriff oder zur Darstellung der Daten
- HSQLDB ist komplett frei und ohne Beschränkungen verfügbar
- Quellcode und eine umfangreiche Dokumentation sind ebenfalls verfügbar[23]

[22] `http://www.jfree.org`
[23] `http://hsqldb.org`

Die HSQLDB wurde in der vorliegenden Arbeit eingesetzt, um die relevanten Daten für die Kodonanpassung der Stand-Alone-Version des Programms „JCat“ in die Applikation zu integrieren.

2.5.5 Java Server Pages (JSP)

„Java Server Pages“ sind ein Framework zur Integration von Java-Code in HTML-Webseiten. Java Server Pages stellen somit eine Erweiterung von einfachen „Java-Servlets“ dar, die alle Java-Klassen bezeichnen, deren Instanzen innerhalb eines Applikationsservers Anfragen von Clients bearbeiten.

Mit der Hilfe von Java Server Pages ist es möglich Webseiten dynamisch zu gestalten und auf diese Weise Daten aus verschiedenen Quellen, wie zum Beispiel Datenbanken, auf der Webseite darzustellen. Der Code für die Anwendung wird dabei direkt in den Quellcode der HTML-Seite eingefügt. Um Java-Code auf einem Webserver auszuführen verwendet man den „Apache Tomcat“ (siehe 2.5.6, Seite 35).

2.5.6 Apache Tomcat

„Apache Tomcat“ ist ein Programm, das eine Umgebung bereitstellt, die es ermöglicht Java-Code auf einem Webserver auszuführen und dynamisch generierte Ergebnisseiten dem anfragenden Benutzer zurückzusenden (siehe Abbildung 2.6). Die Open-Source-Anwendung dient dabei als „Container“, der Java Server Pages und Java Servlets ausführen kann. In Verbindung mit dem integrierten Webserver kann Apache Tomcat direkt als Webserver verwendet werden[24].

In der vorliegenden Arbeit wurde Apache Tomcat sowohl für den Internetauftritt der *Aspergillus*-Datenbank als auch des Programms „JCat“ verwendet.

2.5.7 Simple Object Access Protocol (SOAP)

SOAP ist ein Protokoll zum Austausch von Informationen in einem dezentralisiertem, verteiltem Netzwerk. Es handelt sich dabei um ein XML-basiertes Protokoll das aus drei Teilen besteht:

[24] `http://tomcat.apache.org/`

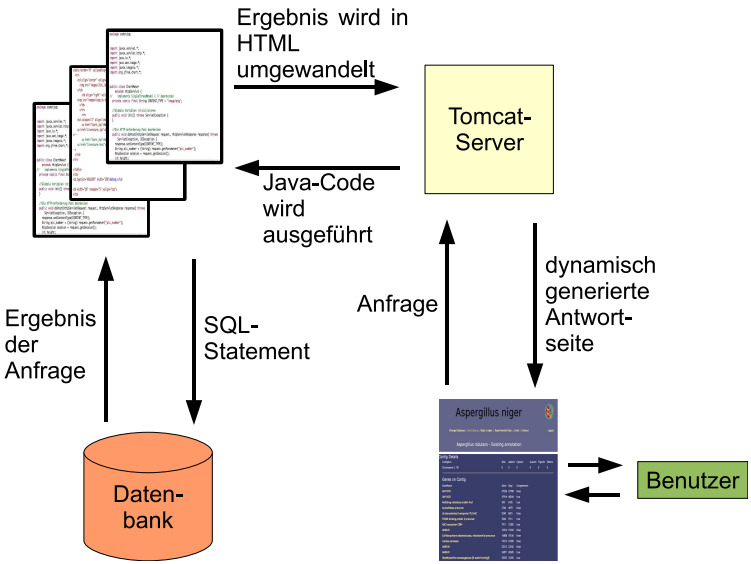

Abbildung 2.6: Dargestellt ist die Interaktion zwischen Tomcat-Server und Datenbankmanagementsystem. Der Benutzer stellt über den Webbrowser eine Anfrage, die an den Tomcat-Server weitergeleitet wird. Dort wird dann ein Java-Skript ausgeführt, das beispielsweise, wie hier dargestellt, eine Anfrage an eine Datenbank stellen kann. Das Ergebnis aus der Anfrage wird anschließend auf dem Tomcat-Server wieder in HTML umgewandelt und zum Aufrufer zurückgeschickt.

1. ein Rahmenwerk, das beschreibt was in der Nachricht enthalten ist und wie die Daten verarbeitet werden müssen

2. eine Reihe von Verschlüsselungsregeln, mit deren Hilfe die Instanzen, der von der Applikation definierten Datentypen, aufgelöst werden können

3. eine Konvention, die festlegt wie entfernte Anfragen verarbeitet und beantwortet werden

SOAP kann in Kombination mit vielen verschiedenen Protokollen verwendet werden. Meist werden Anwendungen aber über das Internet mittels „Hypertext Transfer

Protocol (HTTP)“ miteinander verknüpft[25].

In der vorliegenden Arbeit wurde SOAP dazu verwendet, um dynamisch generierte Stoffwechselwege aus der KEGG-Datenbank auf den Webseiten der *Aspergillus*-Datenbank darzustellen.

[25] `http://www.w3.org/TR/soap/`

3 Ergebnisse und Diskussion

3.1 Die *Aspergillus*-Datenbank „ANigerDB“

Im Teilprojekt B4 des Sonderforschungsbereichs 578 wird die Systembiologie von *Aspergillus niger* untersucht. Ziel ist es ein Modell der Zelle zu entwickeln, mit dessen Hilfe Vorhersagen gemacht werden können, die eine Optimierung der Produktion rekombinanter Proteine erlauben.

Als ersten Schritt in diese Richtung kann man die Entwicklung der *Aspergillus*-Datenbank betrachten. Ursprünglich beinhaltete diese Datenbank nur die Sequenz des *Aspergillus niger* Genoms von Integrated Genomics (siehe 2.2.1.1, Seite 19). Nachdem aber weitere Genome von *A. niger* der Öffentlichkeit zugänglich gemacht wurden, wuchs auch die Anzahl der Genomsequenzen in der Datenbank. Mittlerweile enthält die Datenbank neben den drei verfügbaren Genomsequenzen von *A. niger* auch die Genomsequenzen von weiteren *Aspergillus*-Gattungen wie *A. fumigatus*, *A. nidulans* und *A. oryzae*. Die Genomsequenzen wurden abhängig von der Datenlage mit verschiedenen Methoden annotiert und die Ergebnisse dieser Vorhersagen wurden anschließend auf den Webseiten der Datenbank dargestellt.

3.1.1 Formale Eigenschaften der *Aspergillus*-Datenbank

Die *Aspergillus*-Datenbank verwendet als technischen Rahmen das relationale Datenbankmanagementsystem PostgreSQL. Die Daten werden hier in Tabellen gespeichert. Auf einige Besonderheiten des Datenbankschemas wird im folgenden Abschnitt eingegangen.

Datenbankschema

Die Datenbank besteht aus 39 Tabellen, die über 42 Referenzen miteinander verbunden sind. Das Kerngerüst der Datenbank besteht aus den vier Tabellen „Contig“, „Gene“, „Polypeptide“ und „Protein“. Von diesen Tabellen zweigen sich alle weiteren direkt oder indirekt ab. In Abbildung 3.1 sind diese vier Tabellen mit den wichtigsten Referenzen und betroffenen weiteren Tabellen dargestellt. Die Tabelle „Contig“ enthält Informationen zu den einzelnen Contigs eines Genoms. Die Tabelle

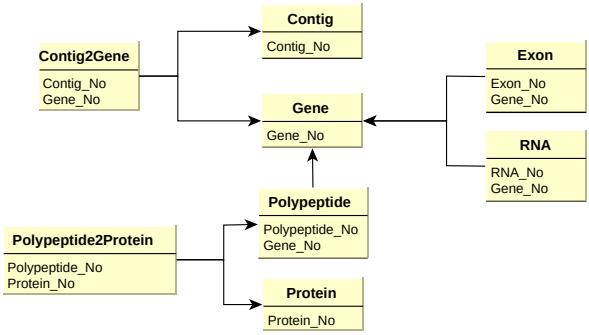

Abbildung 3.1: Dargestellt ist hier das Grundgerüst der *Aspergillus*-Datenbank „ANigerDB“. Neben den Tabellennamen sind noch die jeweiligen primären Schlüssel dargestellt. Die Pfeile stellen Referenzen zwischen den Tabellen dar.

„Contig2Gene“ speichert wo auf dem Contig gefundene Gene liegen. In der Tabelle „Gene“ sind zusätzliche Informationen zu dem Gen gespeichert. Wenn das potentielle Gen ein Intron enthält, werden die entsprechenden Positionen des Exons in der Tabelle „Exon“ abgelegt. Handelt es sich bei dem Gen um einen RNA-kodierenden ORF, so sind Informationen dazu in der Tabelle „RNA“ zu finden. Die Tabelle „Polypeptide“, die direkt mit der Tabelle „Gene“ verbunden ist, enthält die Proteinsequenz des potentiellen Gens. Über die Tabelle „Polypeptide2Protein“ ist diese mit der Tabelle „Protein“ verknüpft. In der Tabelle „Protein“ findet man schließlich Informationen zu dem Protein, das aus dem potentiellem Gen folgt.

Ein weiterer wichtiger Bereich der Datenbank enthält die Informationen, die aus der KEGG-Datenbank integriert wurden (siehe 2.2.2.1, Seite 22). Die wichtigsten Tabellen hierfür sind „EC“, „Reaction“, „Compound“ und „Pathway“ (siehe Abbildung 3.2). Über die Tabelle „EC2Protein“ werden die Daten der KEGG-Datenbank mit den Sequenzdaten der *Aspergillus*-Genome verknüpft. Die Tabelle „EC (Kurzform für ‚Enzyme Classification‘)“ enthält Informationen zu den Enzymen. Über die Tabelle „Reaction2EC“ werden Informationen zu den katalysierten Reaktionen („Reaction“) mit den Enzyminformationen verbunden. In der Tabelle „Compound“ sind Informationen zu den Metaboliten enthalten, die an den Reaktionen als

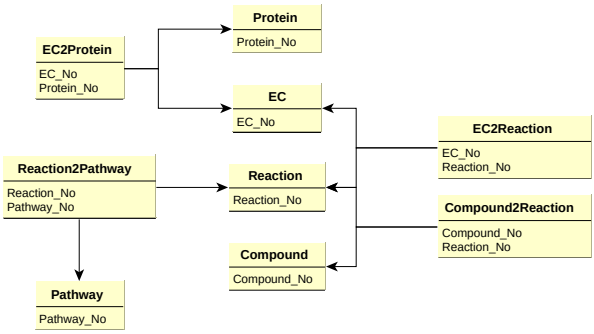

Abbildung 3.2: Dargestellt sind hier die Tabellen der *Aspergillus*-Datenbank „ANigerDB“, die die metabolischen Daten enthalten. Über die Tabelle „Protein“ werden die Tabellen mit den Sequenzdaten der *Aspergillus*-Stämme verknüpft.

Edukte oder Produkte beteiligt sind. Über die Tabelle „Compound2Reaction“ werden sie miteinander verknüpft. In der Tabelle „Pathway“ schließlich sind mehrere Reaktionen zu verschiedenen Stoffwechselwegen zusammengefasst.

Der dritte Bereich des Datenbankschemas, der hier etwas detaillierter vorgestellt werden soll, enthält die experimentellen Daten (siehe Abbildung 3.3). Die expe-

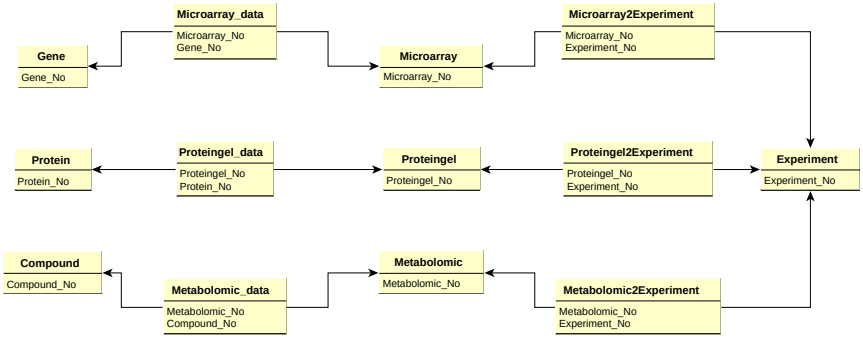

Abbildung 3.3: Dargestellt sind hier die Tabellen der *Aspergillus*-Datenbank „ANigerDB“, die die experimentellen Daten enthalten.

rimentellen Daten lassen sich zunächst in die Bereiche Transkriptom-, Proteom- und Metabolomdaten unterteilen. Zum Speichern allgemeiner Informationen dienen die Tabellen „Microarray“, „Proteingel“ und „Metabolomic“. Um diese speziellen Experimente in einen größeren Kontext stellen zu können, verschiedene Experimente zusammenfassen zu können und allgemeinere Informationen zu dem durchgeführten Experiment speichern zu können, wurde die übergeordnete Tabelle „Experiment“ angelegt. Die untergeordneten Tabellen sind mittels der Verknüpfungstabellen „Microarray2Experiment“, „Proteingel2Experiment“ und „Metabolomic2Experiment“ miteinander verbunden. Die eigentlichen Daten schließlich befinden sich in den Tabellen „Microarray_data“, „Proteingel_data“ und „Metabolomic_data“. Diese Tabellen enthalten jeweils eine Referenz zu dem entsprechenden Experiment, also „Microarray“, „Proteingel“ oder „Metabolomic“ und eine Referenz zu dem beeinflussten Datenbankobjekt „Gene“, „Protein“ oder „Compound“.

3.1.2 Annotation der *Aspergillus niger* Stämme ATCC 1015 und NRRL3 mit verschiedenen Methoden

In der *Aspergillus*-Datenbank befinden sich die Genomsequenzen verschiedener *Aspergillus*-Arten. So weit es möglich war, wurden bereits bestehende Annotationen übernommen. Für die *A. niger*-Stämme ATCC 1015, der vom Joint Genome Institute sequenziert wurde (siehe 2.2.1.2, Seite 20) und NRRL3, der von der Firma Integrated Genomics sequenziert wurde (siehe 2.2.1.1, Seite 19), wurden im Rahmen dieser Arbeit die Sequenzen neu annotiert.

3.1.2.1 Annotation mit Hilfe von BLAST und GlimmerHMM

Die Annotation einer Genomsequenz mit Hilfe von BLAST und GlimmerHMM erfolgt in zwei Etappen:

1. Vorhersage der „Open Reading Frames (ORF)“ aus der Nukleotidsequenz
2. Analyse der ORFs mittels Homologiesuche

Für die Vorhersage von ORFs aus der Sequenz wurde das Programm GlimmerHMM verwendet. Als Modell für die Vorhersage wurde ein Modell gewählt, das mit Genen aus verschiedenen *Aspergillus*-Arten erstellt wurde (siehe 2.1.1, Seite 15).

Die Homologiesuche wurde mittels des Programms BLAST durchgeführt (siehe 2.1.2, 16). Als Referenzdatenbank für diese Suche wurden alle Proteine der Swiss-Prot-TrEMBL-Datenbank verwendet (2.2.2.2, Seite 22). In Abbildung 3.4 ist diese

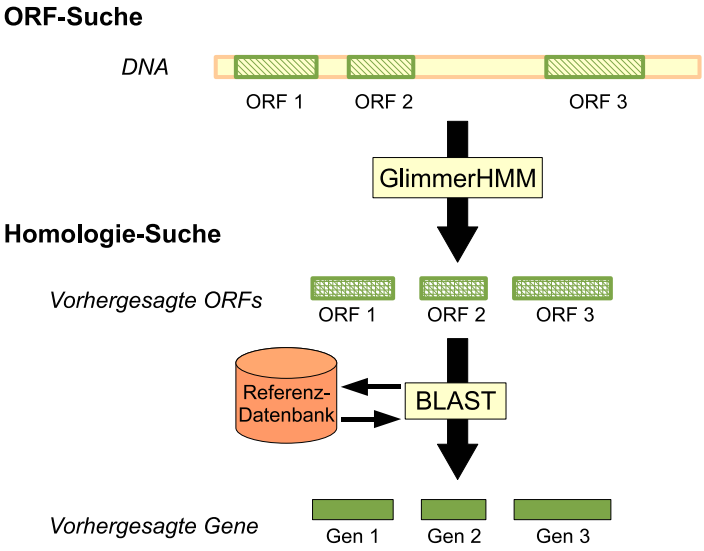

Abbildung 3.4: Dargestellt ist die Vorhersage von Genen in einer Nukleotidsequenz mit Hilfe von BLAST und GlimmerHMM.

Vorgehensweise schematisch dargestellt. Die Verknüpfung der aufeinander folgenden Annotationsschritte, wurde mittels Java-Programmen und der *Aspergillus*-Datenbank realisiert. Es wurden zwei Programme entwickelt, die den jeweiligen Ausgabestrom von GlimmerHMM und BLAST analysieren („parsen“) und die Ergebnisse in die Datenbank schreiben. UML-Diagramme, die diese beiden Programme beschreiben sind im Anhang zu finden.

Nachdem die vorhergesagten ORFs mit BLAST analysiert wurden, wurde eine weitere Analyse der ORFs mit dem Programm tRNAscan-SE durchgeführt, um tRNA-kodierende Gene zu finden, die mittels BLAST nicht gefunden werden können.

3.1.2.2 Annotation mit Hilfe von metaSHARK

Die Genomsequenzen der *A. niger*-Stämme ATCC 1015 und NRRL3 wurden neben der oben beschriebenen Methode zusätzlich mit dem Programmpaket metaSHARK analysiert (siehe 2.1.3, Seite 17). Mit Hilfe des Programms metaSHARK ist es möglich das metabolische Netzwerk aus der Genomsequenz von *A. niger* direkt abzuleiten, da alle notwendigen Programme direkt aus dem Paket heraus aufgerufen werden. Ein Vorteil der Methode ist außerdem, dass aufgrund der Funktionsweise der Analyse auch potentielle Proteine gefunden werden, wenn diese nicht komplett auf der Sequenz liegen. Das kann beispielsweise dann der Fall sein, wenn ein Gen zwischen zwei Contigs liegt. Als Ergebnis wird eine Datei im XML-Format generiert, die mittels eines Java-Programms „geparst“ und in die Datenbank geschrieben wurde. Ein Beispieleintrag aus dieser Datei ist im Anhang zu finden.

3.1.2.3 Integration der Daten aus der KEGG-Datenbank

Um das metabolische Netzwerk für die annotierten Organismen zu erstellen, wurden anschließend noch die Informationen der KEGG-Datenbank in die *Aspergillus*-Datenbank integriert. Die Datenbankstruktur wurde dabei weitgehend von den Dateien der KEGG-Datenbank vorgegeben (siehe Abbildung 3.2, Seite 41). Für die das Auslesen der einzelnen Dateien und die Übertragung in die Datenbank wurde ebenfalls ein Java-Programm entwickelt, dessen UML-Diagramm sich im Anhang befindet. Beispieleinträge der Dateien „ECTable“, „Compound“ und „Reaction befinden sich ebenfalls im Anhang.

3.1.3 Vergleich der Annotationen der *Aspergillus niger*-Stämme

Aufgrund der unterschiedlichen Annotationsmethoden ergeben sich tum Teil größere Unterschiede bei Anzahl und Art der gefundenen Gene und der daraus abgeleiteten Proteine und Enzyme. In Tabelle 3.1 sind die Ergebnisse für die Annotationen der unterschiedlichen *A. niger*-Genome dargestellt. Wie bereits in Abschnitt 3.1.2.2 beschrieben, liefert die Annotation einer DNA-Sequenz mit metaSHARK auch potentielle Proteine, wenn deren Sequenzen unvollständig sind. Diese erhöhte Sensitivität ist auf die besondere Art der Suche zurückzuführen, bei der die Motive der

Tabelle 3.1: Ergebnisse der Annotation von *Aspergillus niger* mit verschiedenen Methoden.

Stamm/ Annotation	gefundene Gene	potentielle Proteine	potentielle tRNAs	distinkte Enzyme
A. niger ATCC1015 (BLAST/GlimmerHMM)	8946	8674	272	809
A. niger ATCC1015 (metaSHARK)	3282	3282	n. v.	1295
A. niger NRRL3 (BLAST/GlimmerHMM)	12668	12441	227	759
A. niger NRRL3 (metaSHARK)	3255	3255	n.v.	1257

Proteine auch in Nukleotidsequenzen gefunden werden können, wenn beispielsweise der vordere oder der hintere Teil der Sequenz nicht vorliegt (siehe 2.1.3, Seite 17). Im Gegensatz dazu liefert die Annotation einer Sequenz mit Hilfe von BLAST und GlimmerHMM nur dann einen Treffer, wenn die gesamte Nukleotidsequenz inklusive Start- und Stopkodon vorliegt. Andererseits erfolgt bei der Analyse einer Sequenz mit metaSHARK nur dann einen Treffer, wenn sich aus der gefundenen Sequenz ein Enzym ableiten lässt. Gene, die für tRNAs kodieren, wurden für die Sequenzen, die mit metaSHARK analysiert wurden, nicht vorhergesagt. In der Anzahl der Treffer drücken sich diese Unterschiede aus. Während die Annotation mit BLAST und GlimmerHMM für die Stämme ATCC 1015 und NRRL3 drei- bis viermal so viele Gene liefert, wird auf Grund der Anzahl, der gefundenen potentiellen Enzyme deutlich, dass metaSHARK für diesen Bereich der Vorhersage wesentlich spezifischere Ergebnisse liefert. Vergleicht man die mit BLAST und GlimmerHMM durchgeführte Annotation der Sequenz des Stamms NRRL3 mit der des Stamms ATCC 1015 fällt auf, dass bei der zuerst genannten Sequenz deutlich mehr Gene lokalisiert werden können. Wenn man jedoch die Anzahl der vorausgesagten distinkten Enzyme aus dieser Sequenz mit in die Betrachtung einbezieht, zeigt sich, dass diese Sequenz nicht unbedingt vollständiger ist. Da die beiden Sequenzen nur vorläufigen Charakter haben, sind Teilbereiche des Genoms vermutlich mehrfach

sequenziert worden. Diese Bereiche sind aus der unfertigen Sequenz noch nicht gelöscht worden und liefern entsprechend mehrfach dieselben vorhergesagten potentiellen Enzyme. Abbildung 3.5 zeigt ein Venn-Diagramm, das die Unterschiede

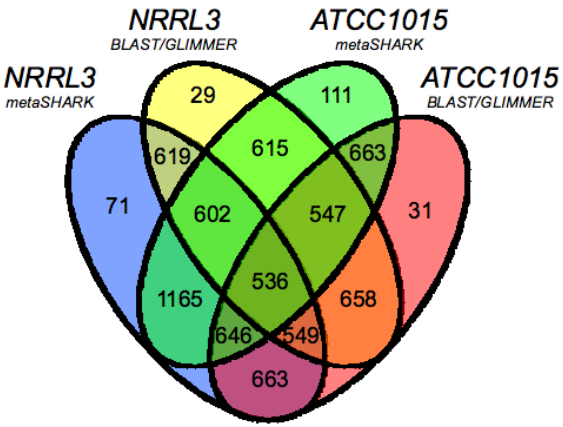

Abbildung 3.5: Dargestellt ist hier ein Venn-Diagramm, das die Verteilung der Enzyme, nach EC-Nummern klassifiziert, für die annotierten *Aspergillus niger* Stämme zeigt. Die Darstellung ist angelehnt an VENNY(Oliveros J.C. (2007) `http://bioinfogp.cnb.csic.es/tools/venny/index.html`).

zwischen den gefundenen Enzymen graphisch darstellt. Ein großer Teil der Enzyme konnte mit beiden Annotationsmethoden in beiden *A. niger*-Stämmen vorhergesagt werden. Die beiden Annotationen mit metaSHARK weisen die größte Menge gemeinsamer vorhergesagter Enzyme auf. Ein Rest von mehr als hundert Enzymen konnte jedoch nur in dem Stamm ATCC 1015 gefunden werden. Ein Vergleich dieser beiden Stämme in der Annotation mit Blast und Glimmer zeigt ebenfalls diese Unterschiede. Der größte Teil, der mit dieser Annotationsmethode vorhergesagten Enzyme, konnte zwar für beide Stämme vorhergesagt werden, aber auch hier bleibt ein Rest von mehr als hundert vorhergesagten Enzymen, die nur in dem Stamm ATCC 1015 gefunden werden konnten. Deutlich werden hier auch noch einmal die zum Teil unterschiedlichen Ergebnisse der beiden Annotationsmethoden. So werden ca. 150 Enzyme, die in der Sequenz des Stamms ATCC 1015, der mit Hilfe von

BLAST und GlimmerHMM annotiert wurde, gefunden, diese aber nicht gefunden, wenn die Sequenz mit metaSHARK annotiert wurde. Auffällig ist in diesem Zusammenhang ebenfalls, dass die Annotationen der *A. niger*-Stämme mit metaSHARK die jeweils größten Mengen exklusiv vorhergesagter Enzyme aufweisen. Während die mit BLAST und GlimmerHMM annotierten Sequenzen hier für beide Stämme nur ca. 30 Enzyme aufweisen, lassen sich bei den mit metaSHARK annotierten Sequenzen für NRRL3 71 Enzyme und für ATCC 1015 111 Enzyme vorhersagen, die jeweils nur in dieser Sequenz gefunden wurden. In Tabelle 3.2 sind beispielhaft fünf Stoffwechselwege und die Anzahl der annotierten Enzyme für die einzelnen *A. niger*-Stämme dargestellt. Auch hier zeigt sich, dass die Annotation mit metaSHARK

Tabelle 3.2: Anzahl gefundener Enzyme für einige beispielhafte Stoffwechselwege verschiedener *Aspergillus niger* Stämme.

Stoffwechselweg (Anzahl aller Enzyme)	ATCC1015 (BLAST/Glimmer)	ATCC1015 (metaSHARK)	NRRL3 (BLAST/Glimmer)	NRRL3 (metaSHARK)
Glykolyse (42)	20	32	24	28
Citratzyklus (25)	14	17	14	17
Atmungskette (36)	14	17	14	18
Glycin-, Serin-, Threonin-Metabolismus (59)	32	41	30	38
Cystein-Metabolismus (21)	10	11	8	11

die meisten Enzyme findet. Die mit BLAST und GlimmerHMM annotierten Stämme zeigen hier nur geringe Unterschiede. Der Stamm ATCC 1015 weist allerdings für die gezeigten Stoffwechselwege, sowohl in der Annotation mit metaSHARK als auch in der Annotation mit BLAST und GlimmerHMM, geringfügig mehr Enzyme auf, als der Stamm NRRL3 (siehe dazu auch Tabelle 3.1, Seite 45).

3.1.4 Ein kurzer Vergleich der unterschiedlichen *Aspergillus*-Arten

Die *Aspergillus*-Datenbank enthält neben den *A. niger*-Genomsequenzen auch die Sequenzen der Genome von *A. fumigatus*, *A. nidulans* und *A. oryzae*. Die einzelnen *Aspergillus*-Arten unterscheiden sich zum Teil erheblich in Verhalten, bevorzugten Umweltbedingungen und metabolischen Eigenschaften (siehe 1.4, Seite 6). Am deutlichsten sollten die Unterschiede der *Aspergillus*-Arten in den annotierten Enzymen erkennbar sein, da diese die Lebensbedingungen des Organismus letztendlich bestimmen. Die Anzahl der vorhergesagten Gene und der potentiellen Enzyme, der noch nicht beschriebenen *Aspergillus*-Arten der Datenbank, sind in Tabelle 3.3 angegeben. Die Annotation wurde dabei für *A. fumigatus*, *A. nidulans* und *A. oryzae* von der jeweiligen Projektgruppe übernommen (siehe 2.2.1, Seite 19). Ein Vergleich der Enzyme der *Aspergillus*-Arten ist in Abbildung 3.6 als Venn-Diagramm dargestellt. Für *A. niger* wurde der Stamm ATCC 1015, der mit BLAST und GlimmerHMM annotiert wurde, als Referenz gewählt. Zu erkennen ist, dass der größte Teil der Enzyme, nämlich 527, in den Nukleotidsequenzen aller vier Arten gefunden werden kann. Die meisten Enzyme, die nur in einer einzigen *Aspergillus*-Art gefunden werden, sind die Enzyme von *A. niger*. Hier finden sich 189 Enzyme, die *A. niger* exklusiv für sich beansprucht. Die anderen drei *Aspergillus*-Arten weisen jeweils nur ca. 20 - 40 exklusive Enzyme auf. Ein ähnliches Bild zeigt sich auch bei dem Vergleich von jeweils einer weiteren *Aspergillus*-Art mit *A. niger*. Hier liegt die Anzahl gemeinsamer Enzyme zwischen 564 und 584. Die Anzahl gemeinsamer Enzyme von jeweils zwei anderen *Aspergillus*-Arten dagegen liegt zwischen 833 und 852. Gründe dafür können neben den bereits erwähnten Unterschieden in den Le-

Tabelle 3.3: Vorhergesagte Gene und potentielle Enzyme der *Aspergillus*-Arten in der *Aspergillus*-Datenbank.

Aspergillus-Art	gefundene Gene	potentielle Enzyme
A. fumigatus	9887	900
A. nidulans	10680	929
A. oryzae	12063	942

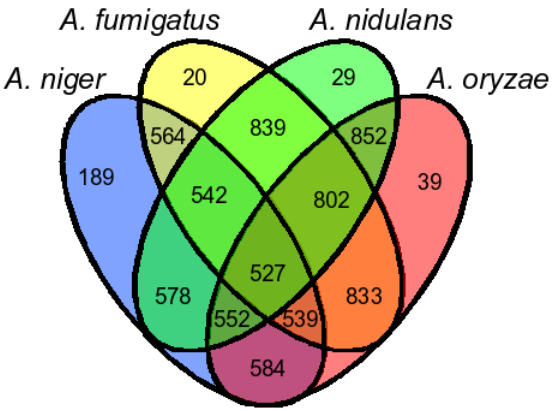

Abbildung 3.6: Dargestellt ist hier ein Venn-Diagramm, das die Verteilung der Enzyme, nach EC-Nummern klassifiziert, für alle *Aspergillus*-Arten der Datenbank zeigt. Als Referenz für *A. niger* wurde der mit Hilfe von BLAST und GlimmerHMM annotierte Stamm ATCC 1015 verwendet.

bensbedingungen und daraus resultierenden Unterschieden im Metabolismus auch in der Methodik der Annotation zu finden sein. So wurden die Genomsequenzen von *A. fumigatus*, *A. nidulans* und *A. oryzae* in enger Kooperation sequenziert, analysiert und schließlich auch gemeinsam publiziert [27, 58, 51]. In Tabelle 3.4 sind die Stoffwechselwege dargestellt, für die in einer oder mehreren *Aspergillus*-Arten keine Enzyme annotiert werden konnten, wohingegen für mindestens eine andere *Aspergillus*-Art der Datenbank mindestens ein Enzym gefunden werden konnte. Auffällig ist, dass anscheinend nur *A. niger* einen Teil, der für den Abbau von Ethylbenzen notwendigen Satz, von Enzymen besitzt. Der weitaus größte Anteil, der Stoffwechselwege ist aber scheinbar in allen *Aspergillus*-Arten, wenigstens rudimentär, vorhanden. Neben den hier dargestellten acht Stoffwechselwegen, werden in der KEGG-Datenbank nämlich mehr als 110 Stoffwechselwege unterschieden, für die in den Sequenzen der *Aspergillus*-Stämme der Datenbank ein oder mehrere Enzyme vorhergesagt werden konnten. Die meisten Stoffwechselwege, die in der Tabelle aufgeführt sind, bestehen jedoch nur aus wenigen Enzymen, so dass die Aussagekraft dieser Unterschiede eher gering ist, denn ein einzelnes Enzym kann

Tabelle 3.4: Stoffwechselwege, für die nicht in allen *Aspergillus*-Arten der *Aspergillus*-Datenbank Enzyme in der jeweiligen Genomsequenz lokalisiert werden konnten. Die Bezeichnung der Stoffwechselwege ist der KEGG-Datenbank entnommen.

Stoffwechselweg	Anzahl der gefundenen Enzyme			
(Anzahl der Gesamtenzyme)	*A. niger*	*A. fumigatus*	*A. nidulans*	*A. oryzae*
Ethylbenzene degradation (6)	4	0	0	0
Biosynthesis of 12-, 14- and 16-membered macrolides (3)	1	0	1	1
Monoterpenoid biosynthesis (17)	0	0	0	1
3-Chloroacrylic acid degradation (2)	2	2	2	0
Tetrachloroethene degradation (3)	1	0	0	0
Puromycin biosynthesis (1)	0	0	0	1
Biosynthesis of vancomycin group antibiotics (1)	1	0	0	0
Bisphenol A degradation (1)	0	0	1	1

unter Umständen bei verschiedenen Methoden der Annotation übersehen werden.

3.1.5 Experimentelle Daten in der Datenbank

Neben den Sequenzdaten und den daraus abgeleiteten Informationen werden in der *Aspergillus*-Datenbank auch experimentelle Daten von *A. niger* verwaltet, die innerhalb des SFB 578 generiert werden. Die experimentellen Daten sind zu diesem Zweck in die Bereiche „Microarray“, „Proteingel“ und „Metabolomic “ unterteilt (siehe 3.1.1, Seite 39). Momentan befinden sich in der Datenbank experimentelle Datensätze zu metabolischen und transkriptionellen Experimenten. Die Experimente sind mit dem *A. niger* Stamm CBS 513.88 verbunden, der im Januar 2007 ver-

öffentlicht wurde [59]. Das hat einen entscheidenden Vorteil, denn die Microarray-Chips, die von dem „MicroArray Department, Universiteit van Amsterdam“[1] stammen, basieren genau auf diesem Stamm. Da die bestehende Annotation für diesen Stamm übernommen wurde, ist auf diese Art eine Verknüpfung zwischen sämtlichen ORFs, die mit dem Chip analysiert werden können, und den Daten der Datenbank möglich. Zum Zeitpunkt der Erstellung der Arbeit befinden sich die in Tabelle 3.5 dargestellten Experimente in der Datenbank. Während auf die Transkriptions-

Tabelle 3.5: Experimentelle Daten der *Aspergillus*-Datenbank. Die Daten stammen aus Experimenten, die am Institut für Mikrobiologie und am Institut für Bioverfahrenstechnik durchgeführt wurden.

BEZEICHNUNG	BESCHREIBUNG
Myzel- vs. Pelletwachstum	4 Microarray Chips (35h Mycel, 35h Pellet, 70h Mycel und 70h Pellet)
Exponentielle vs. stationäre Wachstumsphase	2 Microarray Chips (45h Pellet und 90h Pellet)
Myzel- vs. Pelletwachstum	4 metabolische Datensätze (35h Mycel, 35h Pellet, 70h Mycel and 70h Pellet)
Verschiedene Kultivierungsbedingungen	8 Metabolische Datensätze unterschiedliche Kultivierungsbedingungen bzgl. C-Quelle, pH, fed-batch oder kontinuierliche Kultivierung

daten als Tabelle über die Webseiten der *Aspergillus*-Datenbank zugegriffen werden kann, steht für den Zugriff auf die metabolischen Datensätze auch eine graphische Darstellung zur Verfügung (siehe 3.1.6, Seite 51).

3.1.6 Webinterface der *Aspergillus*-Datenbank

In der *Aspergillus*-Datenbank sind Sequenzdaten von vier verschiedenen *Aspergillus*-Arten gespeichert und zusätzlich die Sequenzdaten von drei verschiedenen

[1] `http://www.micro-array.nl`

Aspergillus niger-Stämmen. Die gesammelten Informationen werden unter der Internetadresse `http://www.jcat.de/ANigerWeb` bereitgestellt. Insgesamt können dort Informationen von sieben unterschiedlichen Systemen abgerufen werden. Primär ist die Seite in drei Bereiche unterteilt:

1. **StartQuery** - Bereich, von dem aus Informationen zu Proteinen, Genen, Contigs oder Stoffwechselwegen abgefragt werden können.
2. **BLAST** - Bereich, in dem die *Aspergillus*-Sequenzen mit Hilfe des BLAST-Algorithmus durchsucht werden können.
3. **Experimental Data** - Bereich, in dem die experimentellen Daten abgerufen werden können.

In Abbildung 3.7 ist die Startseite zu der *Aspergillus*-Datenbank dargestellt. Neben

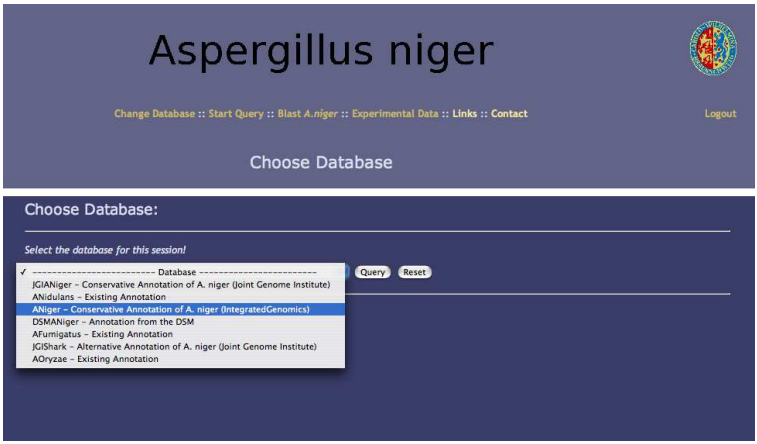

Abbildung 3.7: Dargestellt ist die Startseite der *Aspergillus*-Datenbank. Die Datenbank ist unter der Adresse `http://www.jcat.de/ANigerWeb` verfügbar.

den Nukleotidsequenzen der Contigs und Gene und den Aminosäuresequenzen der vorhergesagten Proteine können alle Sequenzen mittels BLAST-Suche analysiert werden. Die experimentellen Daten der Transkriptionsexperimente werden in tabellarischer Form dargestellt. Die Messergebnisse von metabolischen Experimenten

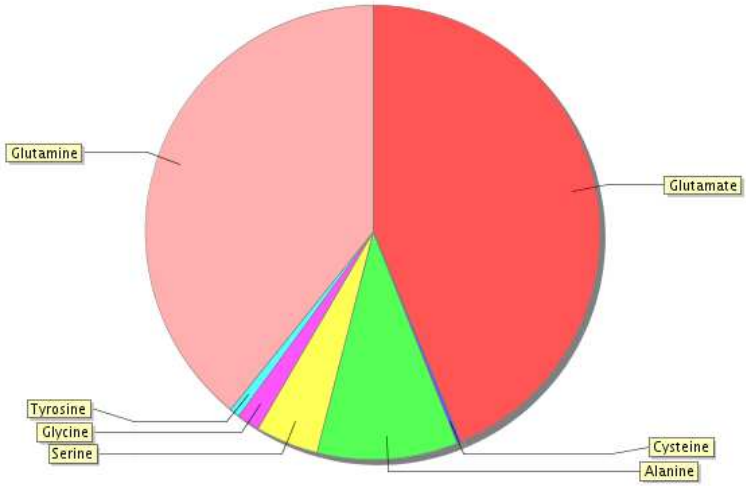

Abbildung 3.8: Dargestellt sind Messungen von Aminosäuren aus einer kontinuierlichen Kultivierung von *A. niger* (550 rpm, pH-Wert 3 auf Glukose). Die Kultivierung erfolgte am Institut für Bioverfahrenstechnik. Metabolite wurden am Institut für Mikrobiologie analysiert.

können, wie bereits erwähnt, auch graphisch dargestellt werden. In Abbildung 3.8 ist als Beispiel eine Messung von verschiedenen Metaboliten gezeigt.

Darstellung von Stoffwechselwegen mit Hilfe der KEGG-Datenbank

Durch eine von der KEGG-Datenbank angebotene SOAP-Schnittstelle ist es möglich Daten zur Laufzeit eines Programms aus der Datenbank abzurufen und weiterzuverarbeiten. Zu diesem Zweck wird einfach ein Java-Paket in das Programmpaket integriert, mit dessen Hilfe dann verschiedene Möglichkeiten zum Zugriff auf die KEGG-Datenbank bereitgestellt werden. Neben der Abfrage von Informationen zu Genen, Proteinen, Metaboliten, etc. besteht die Möglichkeit sich Stoffwechselwege generieren zu lassen, in denen beliebige Enzyme und Metabolite farbig markiert werden. Auf diese Weise ist es möglich für die verschiedenen *Aspergillus*-Arten der Datenbank individuelle Stoffwechselwege darzustellen, je nach dem welche Enzyme in dem jeweiligen Organismus gefunden wurden. In den generierten Stoff-

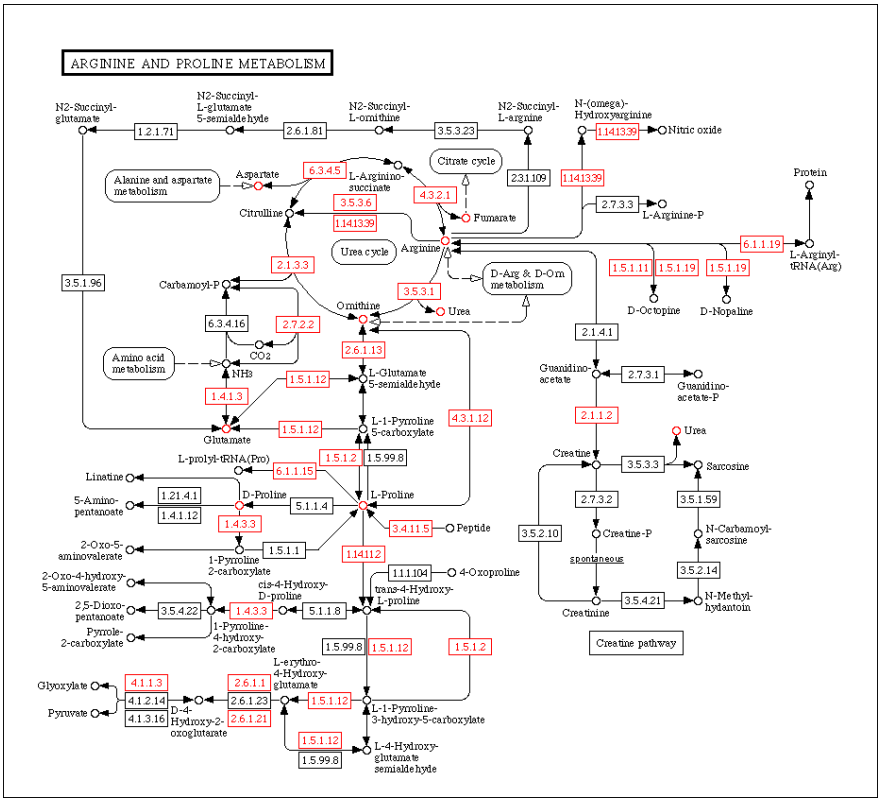

Abbildung 3.9: Dargestellt ist der „Arginin und Prolin Metabolismus“ von *A. niger* ATCC 1015, der mit BLAST und GlimmerHMM annotiert wurde. Für die markierten Enzyme konnten Gene in dem Organismus gefunden werden. Die markierten Metabolite stellen solche dar, die in Experimenten inzwischen identifiziert werden können. Die Karte stammt aus der „Kyoto Encyclopedia of Genes and Genomes“.

wechselwegen werden außerdem die Metabolite farblich hervorgehoben, die bei den metabolischen Experimenten identifiziert werden konnten. In Abbildung 3.9 ist der Stoffwechselweg für den Arginin- und Prolinstoffwechsel als Beispiel dargestellt.

3.1.7 Kurze Zusammenfassung der *Aspergillus*-Datenbank „ANigerDB“

Die *Aspergillus*-Datenbank „ANigerDB “, bietet Genomdaten zu den *Aspergillus*-Arten *A. niger*, *A. fumigatus*, *A. nidulans* und *A. oryzae*. Von den Genomdaten wurden mit Hilfe verschiedener Werkzeuge und Datenbanken die metabolischen Netzwerke abgeleitet. Die „ANigerDB “ wird zur Verarbeitung von Transkriptom-, Proteom- und Metabolomdaten verwendet. Mittels eines benutzerfreundliches Webinterfaces, das unter der Adresse `http://www.jcat.de/ANigerWeb` verfügbar ist, können die Daten abgerufen und gegebenenfalls verändert werden.

3.2 Java Codon Adaptation Tool (JCat)

Im Rahmen dieser Arbeit wurde das, in der Programmiersprache Java geschriebene, Programm „JCat - Java Codon Adaptation Tool“ entwickelt. Das Programm ermöglicht die Anpassung der „codon usage“ einer Gensequenz an ein neues Umfeld, um Proteine mit der Hilfe eines Wirtsorganismus herzustellen. Das Programm kann auf zwei Arten verwendet werden, als

Anwendung im Internet, die unter der Adresse `http://www.jcat.de` zur Verfügung steht und als

eigenständiges Programm, das für Windows und Linux unter der oben angegebenen Adresse heruntergeladen werden kann.

In diesem Abschnitt wird zunächst die Funktionsweise der Kodonadaptierung beschrieben. Weiterhin wird auf spezielle Funktionen eingegangen, die bei der Anpassung der Sequenz optional gewählt werden können. Als letztes werden die beiden Anwendungsoberflächen, auch als „GUIs - Graphical User Interface“ bezeichnet, vorgestellt.

3.2.1 Anpassung der „codon usage“

Als Anpassung der „codon usage“ werden die Schritte bezeichnet, die notwendig sind, um eine fremde Gensequenz, mit für den Wirtsorganismus unüblichen Kodons, an die „codon usage“ im Wirt anzupassen. Dieser Prozess kann in zwei Teile gegliedert werden:

1. Einlesen der Sequenz und gegebenenfalls Übersetzung der Sequenz in eine Aminosäuresequenz.

2. Optimierung der Sequenz und Ausgabe des Ergebnisses.

Im ersten Schritt wird die Sequenz eingelesen. Falls die Sequenz keine Aminosäuresequenz ist, wird sie direkt in eine solche umgewandelt. Im nächsten Schritt wird für jede eingelesene Aminosäure das Kodon im ausgewählten Organismus gesucht,

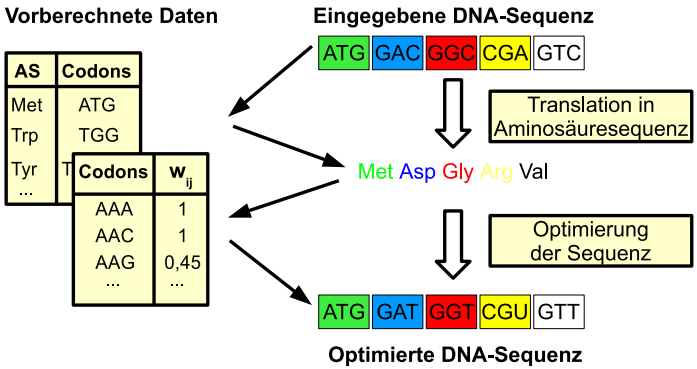

Abbildung 3.10: Dargestellt ist die zweiphasige Anpassung der „codon usage“ einer DNA-Sequenz. Die Translation der DNA-Sequenz und die Anpassung erfolgt mit Hilfe von vorberechneten Daten, die in einer relationalen Datenbank gespeichert sind.

das für die Aminosäure die beste „Relative Adaptiveness (w_{ij})“ aufweist und somit optimal ist. Dieses Kodon wird dann an die neue DNA-Sequenz angehängt. Wenn die gesamte Sequenz durchlaufen wurde, wird die angepasste DNA-Sequenz ausgegeben. In Abbildung 3.10 ist der Anpassungsprozess schematisch dargestellt. Daten, wie zum Beispiel welches Kodon für die entsprechende Aminosäure in dem betrachteten Organismus das optimale ist, sind in einer Datenbank abgelegt, die im Hintergrund des Programms läuft.

3.2.2 Weitere Optionen bei der Anpassung der „codon usage“

Neben der einfachen Anpassung der „codon usage“ bietet das Programm JCat weitere Funktionen, die während des Prozesses berücksichtigt werden können. Die zusätzlichen Optionen dienen vor allem dazu unerwünschte Motive oder Sequenzabschnitte in der optimierten Sequenz zu vermeiden oder bieten erweiterte Funktionen bei der Ein- oder Ausgabe der Sequenz. Im Folgenden werden diese zusätzlichen Funktionen vorgestellt:

3.2.2.1 Der genetische Code der Eingabesequenz

Da der genetische Code bei einigen Organismen und Zellorganellen geringfügig vom Standardcode abweichen kann, wurde in das Programm JCat eine Option integriert, die es ermöglicht auch DNA-Sequenzen mit einem abweichenden Code einzulesen. Insgesamt stehen 17 verschiedene Translationstabellen zur Verfügung, die aus Dateien des NCBI ausgelesen wurden (siehe 2.4.3, Seite 30). Wird ein anderer Code als der Standardcode gewählt, ändert sich an der oben beschriebenen Vorgehensweise der Anpassung im Grunde genommen nichts, außer dass die Translation der eingegebenen DNA-Sequenz in eine Aminosäuresequenz nicht mit Hilfe der Standardcodetabelle erfolgt, sondern mit Hilfe der ausgewählten Tabelle. Die Anpassung der „codon usage“ an den Wirtsorganismus erfolgt dann analog zu der oben beschriebenen Methode.

3.2.2.2 Vermeidung Rho-unabhängiger Transkriptionsterminatoren

Die Vermeidung Rho-unabhängiger Transkriptionsterminatoren spielt eine herausragende Rolle unter den zusätzlichen Optionen des Anpassungsprozesses. Die zufällige Insertion einer solchen Struktur in die optimierte Sequenz führt nämlich zu einem Abbruch der Transkription, so dass kein Protein gebildet wird. Wenn diese Option ausgewählt ist, werden mehrere Schritte in die Anpassung eingefügt:

1. Vorhersage von Transkriptionsterminatoren
2. Wenn ein Treffer erfolgt, Anpassung des entsprechenden Bereichs unter Verwendung von „suboptimalen“ Kodons
3. Erneute Vorhersage von Transkriptionsterminatoren für den entsprechenden Bereich
4. Zwischenspeichern der optimierten Sequenz, die jetzt keinen mehr Transkriptionsterminator enthält
5. Auswahl der optimalen Sequenz unter Verwendung des CAI-Wertes

Transkriptionsterminatoren werden nach der in Abschnitt 2.3.3, Seite 26 beschriebenen Methode vorhergesagt. Wie die Vorhersage erfolgt auch die anschließende

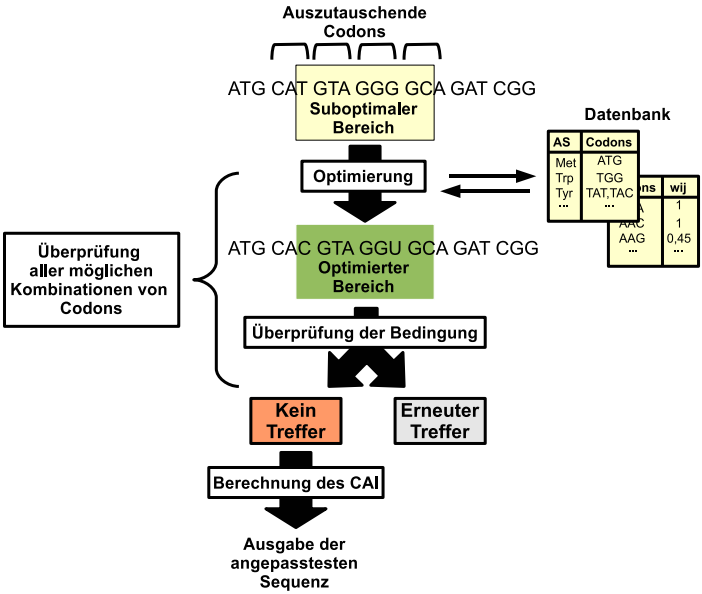

Abbildung 3.11: Dargestellt ist die Eliminierung von unerwünschten Restriktionsenzymbindestellen oder Transkriptionsterminatoren in kodonoptimierten DNA-Sequenzen. Zunächst wird der zu optimierende Bereich festgelegt. Anschließend werden alle Kodons in diesem Bereich mit Kodons ausgetauscht, die für dieselbe Aminosäure kodieren. Wird dann das Muster nicht mehr in der neuen Teilsequenz gefunden, wird diese Sequenz zwischengespeichert. Nachdem alle Kombinationen getestet wurden, wird mit Hilfe des CAI überprüft, welche Sequenz am besten an den Wirtsorganismus angepasst ist. Diese Sequenz wird dann zurückgegeben.

Optimierung, sofern ein Treffer erfolgt, in zwei Teilen. Zunächst wird versucht die „Haarnadelstruktur“ aufzulösen, indem alle Kodons, die im Bereich dieser Struktur liegen, gegen alle anderen möglichen Kodons ausgetauscht werden, die jeweils für dieselbe Aminosäure kodieren müssen. Die so optimierten Teilsequenzen werden erneut daraufhin überprüft, ob sie, laut dem angegebenen Modell, noch eine Haarnadelstruktur ausbilden. Kann keine Haarnadelstruktur mehr gefunden werden, wird diese Teilsequenz gespeichert. Nachdem alle möglichen Kombinationen der Kodons überprüft wurden, wird anschließend mit Hilfe des CAI ermittelt, welche Teilse-

quenz am besten an die „codon usage“ des Wirtsorganismus angepasst ist. In Abbildung 3.11 ist die allgemeine Vorgehensweise der Eliminierung von unerwünschten Sequenzmotiven oder Sequenzmustern dargestellt. Im nächsten Schritt erfolgt dann die Optimierung des Uracil-reichen Bereichs „downstream“ der Haarnadelstruktur. Die Optimierung erfolgt dabei in ähnlicher Weise wie die Eliminierung der Haarnadelstruktur. Alle Kodons in dem fraglichen Bereich werden ausgetauscht und die resultierende Teilsequenz zwischengespeichert, sofern sie nicht mehr dem in Abschnitt 2.3.3 beschriebenen Kriterium für diesen Bereich entspricht. Nachdem auch hier alle möglichen Kombinationen überprüft wurden, entscheidet auch für diese Teilsequenz der CAI darüber, welche Sequenz ausgegeben wird. Nachdem die beiden optimierten Sequenzen miteinander kombiniert wurden und in die Gesamtsequenz integriert wurden, erfolgt die Ausgabe der angepassten Sequenz.

3.2.2.3 Vermeidung prokaryotischer Ribosomenbindestellen (Shine-Dalgarno-Sequenz)

Die Ribosomenbindestelle, die in Prokaryonten auch als Shine-Dalgarno-Sequenz bezeichnet wird, ist ein Sequenzmotiv, das sich zwischen sechs und neun Nukleotide vor dem Startpunkt der Translation befindet. Die Konsensussequenz für *Escherichia coli* lautet: ‚*AGGAGG*‘. In Abbildung 3.12 ist eine schematische Darstellung der Ribosomenbindestelle dargestellt. Um Probleme bei der Translation einer an-

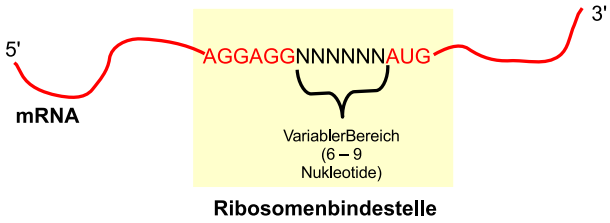

Abbildung 3.12: Schema der prokaryotischen Ribosomenbindestelle.

gepassten Sequenz zu vermeiden und die Synthese von unvollständigen Proteinen zu verhindern, kann in JCat eine Option gewählt werden, die es erlaubt, die angepasste Sequenz auf dieses Motiv zu untersuchen und gegebenenfalls zu verhindern.

Die Optimierung der Sequenz läuft dabei in ähnlicher Weise ab, wie bereits bei der Vermeidung von rho-unabhängigen Transkriptionsfaktoren beschrieben. Zunächst wird überprüft, ob sich ein solches Muster in der angepassten Sequenz befindet. Wenn ein Treffer erfolgt werden die Kodons innerhalb der Konsensussequenz mit anderen Kodons ausgetauscht, die für dieselbe Aminosäure kodieren. Anschließend wird eine erneute Überprüfung der Sequenz durchgeführt. Wird keine Ribosomenbindestelle mehr gefunden, wird diese Sequenz zwischengespeichert. Die am besten an den Wirtsorganismus angepasste Nukleotidsequenz wird auch hier mit Hilfe der Berechnung des CAI für die Sequenz ermittelt. Die am besten angepasste Sequenz, die jetzt keine Ribosomenbindestelle mehr enthält, wird dann zurückgegeben.

3.2.2.4 Vermeidung spezieller Restriktionsenzymbindestellen

Bestimmte Erkennungssequenzen von Restriktionsenzymen sind in der angepassten Sequenz nicht erwünscht, da sie möglicherweise während des „Klonierens“ der Sequenz in die „Multiple Cloning Site“ eines Vektors stören. Zur Vermeidung dieser Sequenzmotive wurden Dateien aus der REBASE-Datenbank, die Informationen zu Restriktionsenzymen enthält, ausgelesen und in das Programm JCat integriert (siehe 2.4.4, Seite 30). Zur Optimierung wird eine einfache Mustersuche mit der Erkennungssequenz des ausgewählten Restriktionsenzyms auf der angepassten Sequenz durchgeführt. Wird die Erkennungssequenz in der angepassten Sequenz gefunden, werden die Kodons in diesem Bereich gegen andere Kodons ausgetauscht, die für dieselbe Aminosäure kodieren. Eine erneute Suche sollte dann kann keine Bindestelle mehr für dieses Restriktionsenzym enthalten. Alle möglichen Kombinationen der Kodons werden getestet und letztendlich wird auch hier die Sequenz zurückgegeben, deren „codon usage“ am besten an den Wirtsorganismus angepasst ist.

3.2.2.5 Unvollständige Anpassung der „codon usage“ der eingegebenen Sequenz

Neben der Anpassung aller Kodons der eingegebenen Sequenz an die „codon usage“ des Wirtsorganismus, gibt es auch die Möglichkeit nur die Kodons anzupassen, die besonders selten in dem Wirtsorganismus verwendet werden. Zu diesem Zweck wurde ein „Cutoff-Value“ der „Relative Adaptiveness (w_{ij})“ für das jeweilige Ko-

don auf ‚0,1‘ festgelegt. Alle Kodons die einen Wert unterhalb des Cutoff-Values aufweisen, werden gegen das jeweils optimale Kodon ausgetauscht. In der Sequenz, die schließlich zurückgeliefert wird, sind die Kodons, die nicht mehr der ursprünglichen Sequenz entsprechen markiert.

Diese Option kann dazu verwendet werden, um eine unangepasste Sequenz, die in einen Wirtsorganismus „kloniert“ werden soll, anzupassen, ohne dass die gesamte Sequenz komplett neu synthetisiert werden muss. Oft sind nur einige sehr seltene Kodons dafür verantwortlich, dass ein Protein in einem Wirtsorganismus nicht synthetisiert wird. Deshalb kann der Austausch einiger Kodons mittels „Site-Directed-Mutagenesis“ die Proteinproduktion unter Umständen bereits signifikant erhöhen.

3.2.3 Berechnung von CAIs aus einer Datei im FASTA-Format

Das Programm JCat bietet außer der Optimierung einer eingegebenen Nukleotidsequenz, auch die Berechnung der CAIs aus einer Liste von Nukleotidsequenzen. Diese Liste kann sowohl in der Internet- als auch in der eigenständigen Programmversion von JCat als Datei im FASTA-Format an das Programm übergeben werden (siehe 2.1.4, Seite 18). Die eigenständige Programmversion von JCat bietet außerdem noch die Möglichkeit Sequenzen im FASTA-Format in ein Fenster zu kopieren. Nachdem die Sequenzen an das Programm übergeben wurden und der Organismus ausgewählt wurde, wird für jede Sequenz der CAI berechnet (siehe 2.3.1, Seite 23). Die Ausgabe der Ergebnisse erfolgt in tabellarischer Form, wobei auf der linken Seite der Bezeichnung der Sequenz angegeben wird und auf der rechten Seite der berechnete CAI.

3.2.4 Das Datenbankmodell von JCat

Daten für die Berechnung der CAIs und zur Kodonanpassung werden innerhalb des Programms JCat in einer Datenbank verwaltet. Für die Internet-Anwendung von JCat wird dabei PostgreSQL (siehe 2.5.1, Seite 31) als Datenbankmanagementsystem verwendet, während in die eigenständige Programmversion von JCat das Javapaket HSQLDB (siehe 2.5.4, Seite 34) integriert wurde.

Das Datenbankschema, das für beide Systeme dasselbe ist, ist in Abbildung 3.13 dargestellt. Die Tabelle „Genome“ enthält die Namen der Organismen, für die die

„codon usage“ berechnet bzw. angepasst werden kann. Außerdem ist in dieser Tabelle angegeben, welche Translationstabelle für diesen Organismus verwendet wird. In den Tabellen „Translation“ und „Codon2Amino_Acid“ sind die verschiedenen Translationstabellen gespeichert. In den Tabellen „Amino_Acid“ und „Codon“ sind Informationen zu den Aminosäuren und die 64 Kodons gespeichert. Die Tabelle „Codon2Genome“ schließlich enthält die „Relative Adaptiveness (w_{ij})“ für die jeweils 64 Kodons aller Organismen.

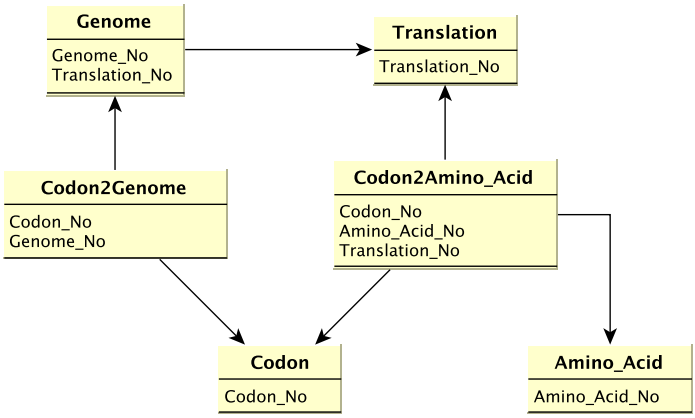

Abbildung 3.13: Dargestellt ist hier das Datenbankschema von JCat, das die, für die Berechnungen benötigten, Werte enthält.

3.2.5 Anwendungsoberflächen von JCat

JCat kann mit zwei verschiedenen Anwendungsoberflächen genutzt werden. Beide Versionen von JCat besitzen eine ähnliche Oberfläche, um zu optimierende Sequenzen einzugeben oder Nukleotidsequenzen zu laden, für die CAIs berechnet werden sollen. Im Folgenden werden die beiden Oberflächen kurz vorgestellt.

3.2.5.1 Der JCat-Webserver

Unter der Adresse `http://www.jcat.de` ist das Programm JCat als Internetanwendung verfügbar. Das Programm unterliegt dabei keinerlei Beschränkungen und kann frei verwendet werden. In Abbildung 3.14 ist die Startseite von JCat dargestellt. Im Durchschnitt werden auf den Webseiten von JCat monatlich mehr als 600 Zugriffe registriert. Neben der Anpassung der „codon usage“ und der Berechnung von CAIs werden auf den Webseiten Anwendungsbeispiele, Literaturhinweise und eine kurze Anleitung zu der Benutzung von JCat angeboten. Momentan kann die „codon usage“ von 377 Prokaryonten und 9 Eukaryonten angepasst werden. Für dieselbe Anzahl können gleichzeitig auch CAIs berechnet werden.

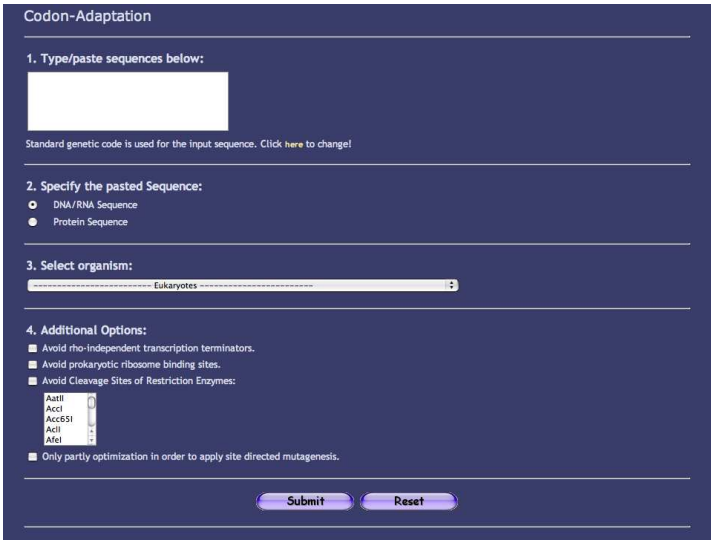

Abbildung 3.14: Startseite der Webseiten von JCat. JCat ist unter der Adresse `http://www.jcat.de` verfügbar.

3.2.5.2 JCat als eigenständiges Programm

JCat kann unter der Adresse `http://www.jcat.de/Download.jsp` als eigenständiges Programm heruntergeladen werden. Dort steht jeweils eine Version für die Betriebsysteme Windows und Linux zur Verfügung. Die Installation ist unkompliziert, da durch die Verwendung von ‚HSQLDB' kein zusätzliches Datenbanksystem installiert werden muss. Diese Version von JCat besitzt den gleichen Funktionsumfang wie die Internetanwendung, allerdings enthält diese Version nicht die zusätzlichen Literaturhinweise, Anwendungsbeispiele, etc. Der große Vorteil dieser Version ist jedoch, dass zum Berechnen von CAIs und Optimieren von Sequenzen keine Verbindung zum Internet nötig ist. Seit dem Erscheinen von JCat als eigenständigem Programm im Oktober 2007 bis Mitte November 2007 wurde JCat mehr als 70 mal heruntergeladen.

3.2.6 Anwendungsbeispiele für JCat aus der Literatur

Das Programm JCat konnte inzwischen erfolgreich bei verschiedenen Projekten eingesetzt werden. So konnte beispielsweise die Synthese von verschiedenen Proteinen in *Bacillus megaterium* durch die Anpassung der „codon usage" mittels JCat erheblich verbessert oder zum Teil so gar erst ermöglicht werden [11, 73, 80, 79]. Weiterhin war JCat hilfreich bei der Entwicklung einer Strategie für die Verbesserung der Translation von GC-reichen DNA-Sequenzen aus *Rhodopseudomonas palustris* in *Escherichia coli* [9]. Auch bei der Visualisierung theoretischer Expressivität von Genen bzw. der Menge resultierendem Protein, wurden die entwickelten Java-Klassen von JCat verwendet, um virtuelle 2-D-Gele zu visualisieren [33].

JCat hat sich inzwischen als Werkzeug zur Anpassung der „codon usage" etabliert und wird bei neuen Projekten mit ähnlichem Schwerpunkt häufig zitiert [61, 76, 77, 78, 4].

3.2.7 Anpassung des *Escherichia coli* Arsenat Reduktase Gens *arsC* an die „codon usage" von *Bacillus megaterium*

Um die Funktionsweise von JCat zu illustrieren, wird in diesem Abschnitt das Gen „*arsC*" aus *E. coli*, das für eine „Arsenat-Reduktase" kodiert, an die „codon usa-

ge“ von *B. megaterium* angepasst. In dem Beispiel wird eine unvollständige Anpassung der „codon usage“ der eingegebenen Sequenz vorgenommen (siehe 3.2.2, Seite 57). Abbildung 3.15 zeigt die graphische Ausgabe von JCat. Der CAI der Eingabesequenz beträgt 0,27. Nach der Anpassung der schlechtesten Kodons ergibt sich ein CAI von 0,56. Die durchschnittliche „codon usage“ für *B. megaterium* beträgt 0,44. Die durchschnittliche „codon usage“ berechnet sich als das arithmetische Mittel der CAIs aller vorhergesagten Gene eines Organismus. Neben dem CAI wird auch der GC-Gehalt für die eingegebene und angepasste Sequenz berechnet. Während die eingegebene Sequenz einen GC-Gehalt von 52,2% aufweist, hat die optimierte Sequenz einen GC-Gehalt von 43,5% und liegt damit deutlich näher am Durchschnitt des Organismus, der bei 38,3% liegt. Das Alignment der Originalsequenz von „*arsC*“ aus *E. coli* mit der an die „codon usage“ von *B. megaterium* angepassten Sequenz in Alignment 3.1 zeigt, dass 24 Kodons in der Sequenz geändert wurden. Das bedeutet, dass in der eingegebenen Sequenz 24 Kodons, die *E. coli* in der Gensequenz von „*arsC*“ verwendet, in *B. megaterium* eine „Relative Adaptiveness (w_{ij})“ von weniger als 0,1 aufweisen.

3.2.8 Kurze Zusammenfassung des Programms „JCat“

Das Programm „JCat“ ermöglicht die Anpassung der „codon usage“ eines Zielgens an die „codon usage“ des Produktionswirtes. Hierfür nutzt es den etablierten „Codon Adaptation Index (CAI)“ zur Bewertung der Kodons. Während des Anpassungsprozesses können verschiedene Optionen wie unerwünschte rho-unabhängige Transkriptionsterminatoren, Ribosomenbindestellen und Erkennungssequenzen von Restriktionsendonukleasen berücksichtigt werden. Das Programm JCat steht unter der Adresse `http://www.jcat.de` als *stand alone*- oder als *online*-Version zur Verfügung.

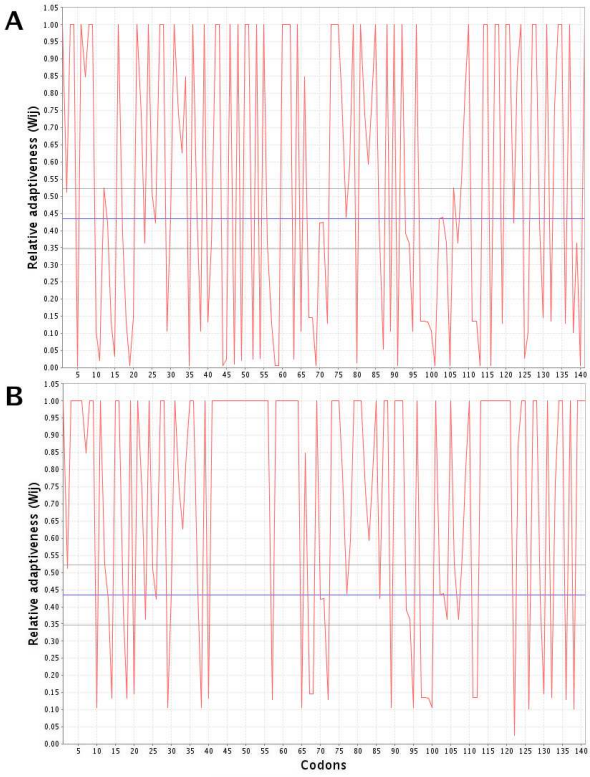

Abbildung 3.15: Dargestellt ist hier die unvollständige Anpassung der „codon usage“ von ‚*arsC*‘ aus *Escherichia coli* an die „codon usage“ von *Bacillus megaterium*. Oben in der Abbildung ist die „codon usage“ für die eingegebene Sequenz in *B. megaterium* aufgetragen (A). Der untere Teil der Abbildung zeigt die „codon usage“ für die angepasste Sequenz (B). Auf der x-Achse ist die „Relative Adaptiveness (w_{ij})“ aufgetragen. Die y-Achse gibt die Nummer des jeweiligen Kodons in der Sequenz an. Die horizontalen Linien im unteren Bereich der Abbildungen zeigen die durchschnittliche „codon usage“ für *B. megaterium* mit den Standardabweichungen an.

```
Original        ATGAGCAACATTACCATTTATCACAACCCGGCCTGCGGCACGTCGCGTAATACGCTGGAG 60
Angepasst       ATGAGCAACATTACAATTTATCACAACCCGGCTTGCGGCACGTCTCGTAATACGTTAGAG 60
                ************** ***************** *********** ********* * ***

Original        ATGATCCGCAACAGCGGCACAGAACCGACTATTATCCATTATCTGGAAACTCCGCCAACG 120
Angepasst       ATGATCCGCAACAGCGGCACAGAACCGACTATTATCCATTATTTAGAAACTCCGCCAACG 120
                ******************************************* * ***************

Original        CGCGATGAACTGGTCAAACTCATTGCCGATATGGGGATTTCCGTACGCGCGCTGCTGCGT 180
Angepasst       CGCGATGAATTAGTAAAATTAATTGCTGATATGGGTATTTCTGTACGCGCGTTATTACGT 180
                ********* * ** *** * ***** ******** ***** ********* *  * ***

Original        AAAAACGTCGAACCGTATGAGGAGCTGGGCCTTGCGGAAGATAAATTTACTGACGATCGG 240
Angepasst       AAAAACGTAGAACCGTATGAGGAGTTAGGCCTTGCGGAAGATAAATTTACTGACGATCGT 240
                ******** *************** * ********************************

Original        TTAATCGACTTTATGCTTCAGCACCCGATTCTGATTAATCGCCCGATTGTGGTGACGCCG 300
Angepasst       TTAATCGACTTTATGCTTCAACACCCGATTTTAATTAATCGCCCGATTGTGGTGACGCCG 300
                ******************** ********* * ***************************

Original        CTGGGAACTCGCCTGTGCCGCCCTTCAGAAGTGGTGCTGGAAATTCTGCCAGATGCGCAA 360
Angepasst       TTAGGAACTCGCTTATGCCGCCCTTCAGAAGTGGTGTTAGAAATTTTACCAGATGCGCAA 360
                 * ********* * ********************* * ****** * *************

Original        AAAGGCGCATTCTCCAAGGAAGATGGCGAGAAAGTGGTTGATGAAGCGGGTAAGCGCCTG 420
Angepasst       AAAGGCGCATTCTCTAAGGAAGATGGCGAGAAAGTGGTTGATGAAGCGGGTAAGCGCTTA 420
                ************** ******************************************* *

Original        AAA 423
Angepasst       AAA 423
                ***
```

Alignment 3.1: Dargestellt ist hier das Alignment zwischen der originalen Gensequenz von „*arsC*“ aus *Escherichia coli* (Original) und die ausgegebene Sequenz von JCat, nachdem sie mittels unvollständiger Anpassung an die „codon usage“ von *Bacillus megaterium* angepasst wurde (Angepasst).

4 Ausblick

Da die Arbeiten an bioinformatischen Projekten aufgrund von aktualisierten Daten, neueren Programmbibliotheken, neuen Techniken und nicht zuletzt aufgrund von Anforderungen, die Benutzer während des laufenden Betriebes an die Entwickler stellen, eigentlich nie zum Abschluss kommen, sind sowohl für die *Aspergillus*-Datenbank als auch für das Programm JCat noch verschiedene Aufgaben offen geblieben:

Aspergillus-Datenbank

- Neben den in dieser Arbeit für die Datenbank verwendeten Genomsequenzen verschiedener *Aspergillus*-Arten, stehen noch weitere Genomprojekte kurz vor dem Abschluss (*Aspergillus flavus*, *Aspergillus terreus*, *Aspergillus fischeri*, *Aspergillus clavatus*). Diese Sequenzen sollten zur Vervollständigung der Daten ebenfalls in die *Aspergillus*-Datenbank integriert werden.

- Das Ziel des Teilprojektes B4, nämlich Vorhersagen über den Zustand der Zelle zu verschiedenen Zeitpunkten der Kultivierung von *A. niger* machen zu können, muss weiterverfolgt werden, indem beispielsweise weitere experimentelle Datenreihen aufgenommen werden. Gerade im Bereich der Transkriptom- und Proteomexperimente besteht dort noch Bedarf.

- Zur Vorhersage von Prozessen in der Zelle müssen verschiedene Konzepte evaluiert und gegebenenfalls an die Bedingungen von *A. niger* angepasst werden. Erste Schritte wurden in diesem Zusammenhang schon in Form von metabolischen Flussanalysen, die aus stöchiometrischen Modellen von *A. niger*, die mit Hilfe der Datenbank abgeleitet wurden, innerhalb des Teilprojektes B4 unternommen [55].

- Um die Datenlage aktuell zu halten sind Aktualisierungen der Daten notwendig. Das betrifft beispielsweise die Daten, die aus der KEGG-Datenbank stammen oder auch die Sequenzdaten aus den verschiedenen Datenquellen, da diese zum Teil weiterhin verbessert werden. Auch neue Eigenschaften in

der Darstellung sind erwünscht, wie beispielsweise eine verbesserte Darstellung der „Contigs“, die momentan nur in textbasierter Form möglich ist.

JCat

- Die ständige Zunahme sequenzierter Organismen bedingt, dass auch JCat aktualisiert werden muss. Wenngleich die neu sequenzierten Organismen in den allermeisten Fällen nicht sofort als Wirtsorganismen für die heterologe Proteinexpression erschlossen werden, so ist die Berechnung von CAIs für diese Organismen eine wertvolle Funktion von JCat.

- Weitere Optionen bei der Anpassung der „codon usage“ sind denkbar. So ist beispielsweise die Kozak-Sequenz, die Ribosomenbindestelle auf der mRNA von Eukaryonten, nicht so konserviert wie die Shine-Dalgarno-Sequenz. Jedoch ließe sich auch diese Sequenz vorhersagen und könnte somit ebenfalls in der angepassten Sequenz, falls erwünscht, vermieden werden.

5 Zusammenfassung

Der Braunschweiger SFB 578 „Vom Gen zum Produkt“ befasst sich mit der Erfassung aller biologischen und verfahrenstechnischen Parameter, die dem Produktionsprozess rekombinanter Proteine in dem filamentösem Pilz *Aspergillus niger* zu Grunde liegen. Gewonnene Erkenntnisse sollen zu systembiologischen Modellen verarbeitet werden und über Prognosen zur Optimierung des Gesamtprozesses dienen. Um eine genaue Momentaufnahme der Vorgänge in der Zelle zu erhalten, werden innerhalb des Teilprojektes B4 „Systembiologie der Produkt- und Pelletbildung durch *Aspergillus niger*“ Transkriptom-, Proteom- und Metabolomdaten aufgenommen. In der vorliegenden Arbeit wurde eine nutzerfreundliche, netzwerkbasierte Computerschnittstelle geschaffen, mit der es möglich ist diese Daten zu speichern, abzurufen und falls nötig zu verändern. Das System, das aus einer Kombination von Datenbankmanagementsystem und Webserver besteht, ist unter der Adresse `http://www.jcat.de/ANigerWeb` verfügbar. Um vergleichende Analysen zwischen verschiedenen *Aspergillus*-Arten durchführen zu können, wurden alle zum Zeitpunkt der Fertigstellung der Arbeit frei verfügbaren Genom- und abgeleiteten Daten von *Aspergillus*-Arten (*A. niger*, *A. fumigatus*, *A. nidulans* und *A. oryzae*) in die Datenbank aufgenommen. Der besondere Fokus auf *A. niger* spiegelt sich auch in der Datenbank wieder, da dort die Genomsequenzen von drei verschiedenen Stämmen dieser Art gespeichert sind. Neben der Genomsequenz, die von der amerikanischen Firma Integrated Genomics erworben wurde, wurden zwei zusätzliche Genomsequenzen von *A. niger* während der Anfertigung dieser Arbeit veröffentlicht. Die Sequenzen stammen vom „Joint Genome Institute“ und von der niederländischen Firma „DSM“.

Eine computergestützte Optimierung eines wichtigen, den Proteinproduktionsprozess beeinflussenden Faktors, der Kodonnutzung („codon usage“), wurde in einem zweiten Teil der Arbeit mit dem Programm JCat implementiert. Das Programm ermöglicht auf einfache Art und Weise die Anpassung der Kodonnutzung eines Zielgens an die Kodonnutzung eines Wirtsorganismus für eine optimale Produktion. Gerade die im Fokus des SFB 578 stehenden Organismen *A. niger* und *Bacillus megaterium* wurden dabei berücksichtigt. Das Werkzeug knüpft dabei an bereits etablierte bioinformatische Konzepte zur Bewertung von Kodons an und schließt damit die

Lücke, die zwischen Analyse und Anwendung gewonnener Erkenntnisse besteht. Neben einer simplen Anpassung der Kodonnutzung einer eingegebenen Sequenz an die Bedingungen in einem Wirtsorganismus wurden verschiedene Funktionen in das Programm integriert, um die Wahrscheinlichkeit einer erfolgreichen Proteinproduktion zu erhöhen. Das Programm kann unter der Adresse `http://www.jcat.de` direkt verwendet werden oder von dort heruntergeladen und dann lokal auf dem Rechner installiert werden. Das Programm wurde bereits für eine Reihe publizierter experimenteller Optimierungen von Proteinproduktionsprozessen erfolgreich angewendet und von der Firma Sanofi-Aventis lizensiert.

6 Anhang

UML-Diagramme von Java-Programmen

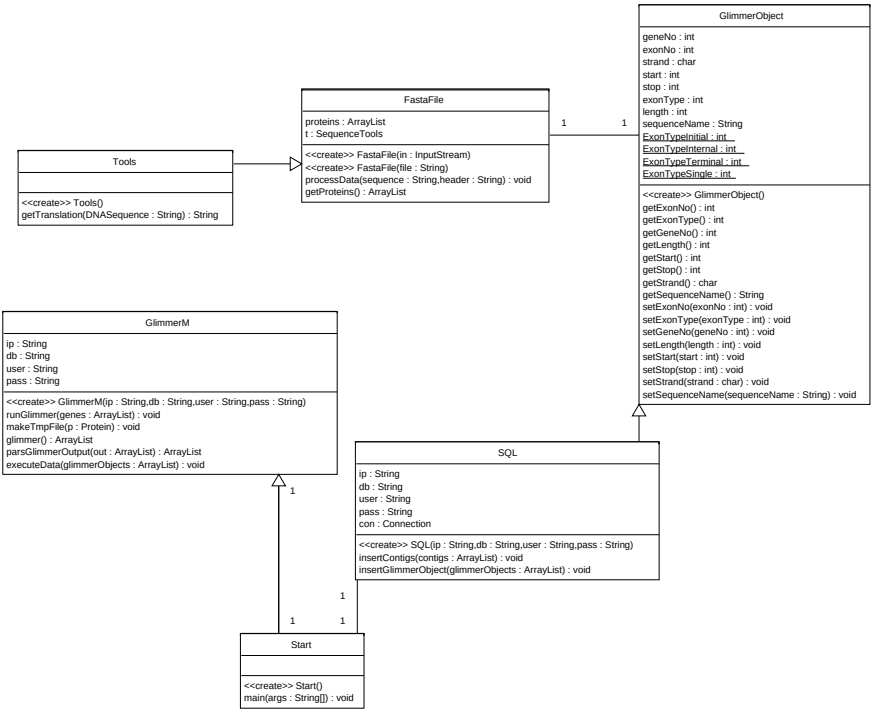

Abbildung 6.1: UML-Diagramm des Java-Programms zur Verarbeitung des Ausgabestroms von GlimmerHMM.

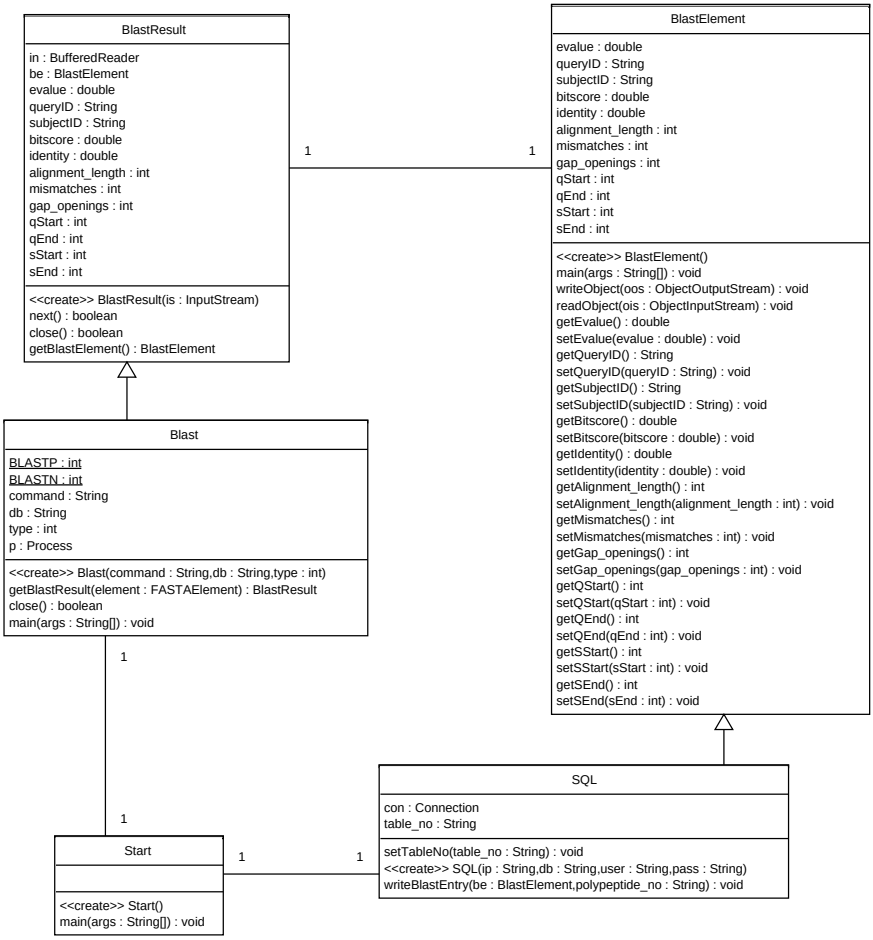

Abbildung 6.2: UML-Diagramm des Java-Programms zur Verarbeitung des Ausgabestroms von BLAST.

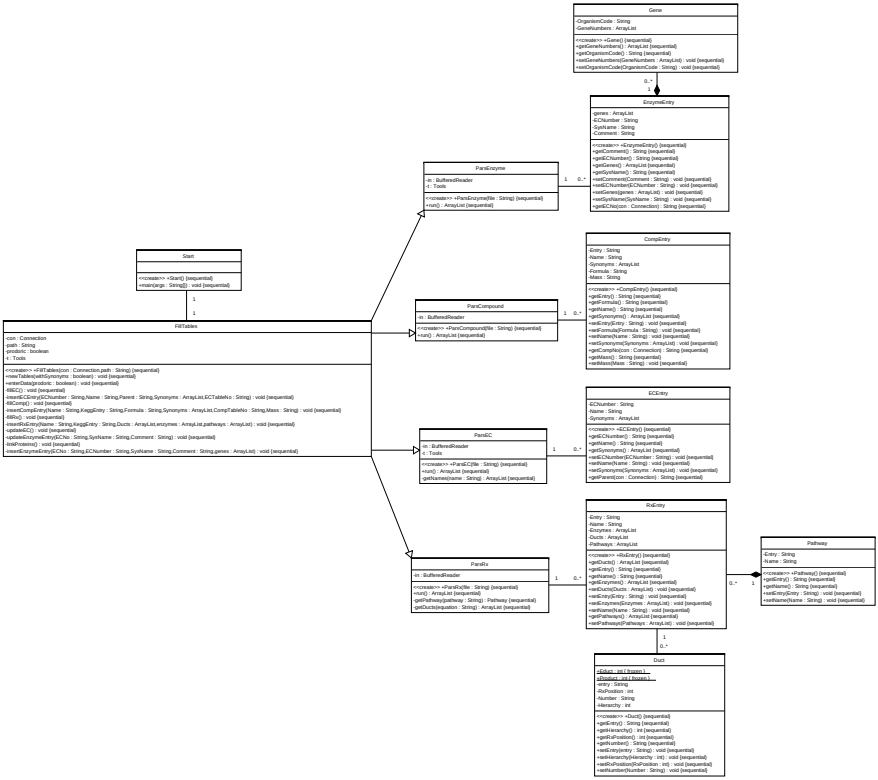

Abbildung 6.3: UML-Diagramm des Java-Programms zum Einfügen der Daten aus der KEGG-Datenbank.

Beispieleinträge aus verwendeten Dateien

```
<?xml version="1.0" encoding="UTF-8"?>
<shark>
  <enzyme id="1.1.1.1">
    <profile id="7p1.1.1.1">
      <hit id="JGINiger|7p1.1.1.1|1" source="shark">
        <e-value>5.4E-60</e-value>
        <sequence>scaffold_4</sequence>
        <strand>+</strand>
        <start>862069</start>
        <end>872489</end>
        <CDS>862069..862283</CDS>
        <CDS>862331..862389</CDS>
        <CDS>862437..863036</CDS>
        <CDS>863073..863353</CDS>
        <CDS>872151..872185</CDS>
        <CDS>872471..872489</CDS>
        <cDNA>TTTGAAATGGCTGCCTCCAGCATCCGATTCGGCCCTGGAGCCACGAAAG...</cDNA>
        <translation>FEMAASSIRFGPGATKEVGMDFANMKAKRVCIVTDE...</translation>
      </hit>
      ...
    </profile>
    ...
  </enzyme>
  ...
</shark>
```

Listing 6.1: Dargestellt ist hier ein einzelner Eintrag der XML-Datei, die das Programm metaSHARK als Ausgabe erzeugt.

```
+D
#<H1>Enzyme EC numbers</H1>
#<H3>EC (Enzyme Commission) numbers assigned by
#<A HREF="http://www.chem.qmw.ac.uk/iupac/jcbn/">IUPAC-IUBMB</A><H3>
#
A<B>@1. Oxidoreductases;@</B>
B  @1.1@  Acting on the CH-OH group of donors;
C    @1.1.1@  With NAD+ or NADP+ as acceptor
D      1.1.1.1  alcohol dehydrogenase; aldehyde reductase; ADH; ...
D      1.1.1.2  alcohol dehydrogenase (NADP+); aldehyde reductase...
...
```

Listing 6.2: Dargestellt ist ein Auszug der Datei „ECTable“ aus der KEGG-Datenbank.

```
///
ENTRY       C05706                        Compound
NAME        Se-Propenylselenocysteine se-oxide
FORMULA     C6H11NO3Se
MASS        224.9904
REACTION    R04934
PATHWAY     PATH: map00450  Selenoamino acid metabolism
DBLINKS     PubChem: 8013
ATOM        11
            1   Z   Se    -1.5207    -0.6724
            2   C1b C      0.2759     0.3724
            3   C2b C     -3.3138     0.4966
...
BOND        10
            1     1   2 1
            2     1   3 1
            3     1   4 2
...
///
```

Listing 6.3: Dargestellt ist ein Auszug der Datei „Compound“ aus der KEGG-Datenbank

```
///
ENTRY       R00002                      Reaction
NAME        Reduced ferredoxin:dinitrogen oxidoreductase
            (ATP-hydrolysing)
DEFINITION  16 ATP + 16 H2O <=> 8 e- + 8 H+ + 16 Orthophosphate + 16 ADP
EQUATION    16 C00002 + 16 C00001 <=> 8 C05359 + 8 C00080 + 16 C00009 +
            16 C00008
RPAIR       RP: A00003  C00002_C00008 main
            RP: A00010  C00002_C00009 main
            RP: A05676  C00001_C00009 leave
ENZYME      1.18.6.1
///
```

Listing 6.4: Dargestellt ist ein Auszug der Datei „Reaction“ aus der KEGG-Datenbank

7 Abkürzungsverzeichnis

ANSI	American National Standards Institute
API	Application Programming Interface
BLAST	Basic Local Alignment Search Tool
CAI	Codon Adaptation Index
Contig	Contiguous DNA-Sequence
DNA	Desoxyribonukleinsäure
EC	Enzyme Classification
EBI	European Bioinformatic Institute
EMBL	European Molecular Biology Laboratory
FASTA	Fast-All
GC/MS	Gas chromatography-mass spectrometry
GNU	GNU's Not Unix
GUI	Graphical User Interface
HMM	Hidden Markov Model
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JCat	Java Codon Adaptation Tool
JDBC	Java Database Connectivity
JSP	Java Server Pages
KEGG	Kyoto Encyclopedia of Genes and Genomes
KNN	Künstliche neuronale Netze
LC/MS	Liquid chromatography-mass spectrometry
mRNA	Messenger-Ribonukleinsäure
NCBI	National Center for Biotechnology Information
ORF	Open Reading Frame
REBASE	Restriction Enzyme Database
RNA	Ribonukleinsäure
rpm	Revolutions Per Minute
SFB	Sonderforschungsbereich
SOAP	Simple Object Access Protocol
SVM	Suppport Vector Machines

SQL	Structured Query Language
TIGR	The Institute for Genomic Research
UML	Unified Modeling Language
tRNA	Transfer-Ribonukleinsäure
XML	Extensible Markup Language

8 Literaturverzeichnis

[1] Alcamo, I. E. (1994). *Fundamentals of Microbiology - Fourth Edition*. The Benjamin/Cummings Publishing Company, Inc.

[2] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol,* **215** (3), 403–410.

[3] Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W. & Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res,* **25** (17), 3389–3402.

[4] Angellotti, M. C., Bhuiyan, S. B., Chen, G. & Wan, X.-F. (2007). CodonO: codon usage bias analysis within and across genomes. *Nucleic Acids Res,* **35** (Web Server issue), W132–W136.

[5] Bairoch, A., Apweiler, R., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., Martin, M. J., Natale, D. A., O'Donovan, C., Redaschi, N. & Yeh, L.-S. L. (2005). The universal protein resource (uniprot). *Nucleic Acids Res,* **33** (Database issue), D154–D159.

[6] Baker, S. E. (2006). *aspergillus niger* genomics: past, present and into the future. *Med Mycol,* **44 Suppl 1**, S17–S21.

[7] Belacel, N., Wang, Q. & Cuperlovic-Culf, M. (2006). Clustering methods for microarray gene expression data. *OMICS,* **10** (4), 507–531.

[8] Bennetzen, J. L. & Hall, B. D. (1982). Codon selection in yeast. *J Biol Chem,* **257** (6), 3026–3031.

[9] Bernstein, J. R., Bulter, T., Shen, C. R. & Liao, J. C. (2007). Directed evolution of ribosomal protein S1 for enhanced translational efficiency of high GC *Rhodopseudomonas palustris* DNA in *Escherichia coli*. *J Biol Chem,* **282** (26), 18929–18936.

[10] Berth, M., Moser, F. M., Kolbe, M. & Bernhardt, J. (2007). The state of the art in the analysis of two-dimensional gel electrophoresis images. *Appl Microbiol Biotechnol,* **76** (6), 1223–1243.

[11] Biedendieck, R. (2006). *Bacillus megaterium: Versatile Tools for Production, Secretion and Purification of Recombinant Proteins*. Doktorarbeit, Technische Universität Braunschweig Institut für Mikrobiologie.

[12] Bieri, T., Blasiar, D., Ozersky, P., Antoshechkin, I., Bastiani, C., Canaran, P., Chan, J., Chen, N., Chen, W. J., Davis, P., Fiedler, T. J., Girard, L., Han, M., Harris, T. W., Kishore, R., Lee, R., McKay, S., Müller, H.-M., Nakamura, C., Petcherski, A., Rangarajan, A., Rogers, A., Schindelman, G., Schwarz, E. M., Spooner, W., Tuli, M. A., Auken, K. V., Wang, D., Wang, X., Williams, G., Durbin, R., Stein, L. D., Sternberg, P. W. & Spieth, J. (2007). WormBase: new content and better access. *Nucleic Acids Res,* **35** (Database issue), D506–D510.

[13] Birney, E., Clamp, M. & Durbin, R. (2004). GeneWise and Genomewise. *Genome Res,* **14** (5), 988–995.

[14] Brinkmann, U., Mattes, R. E. & Buckel, P. (1989). High-level expression of recombinant genes in Escherichia coli is dependent on the availability of the *dnaY* gene product. *Gene,* **85** (1), 109–114.

[15] Calderone, T. L., Stevens, R. D. & Oas, T. G. (1996). High-level misincorporation of lysine for arginine at AGA codons in a fusion protein expressed in *Escherichia coli*. *J Mol Biol,* **262** (4), 407–412.

[16] Carbone, A., Zinovyev, A. & Képès, F. (2003). Codon adaptation index as a measure of dominating codon bias. *Bioinformatics,* **19** (16), 2005–2015.

[17] Carter, M. G., Piao, Y., Dudekula, D. B., Qian, Y., VanBuren, V., Sharov, A. A., Tanaka, T. S., Martin, P. R., Bassey, U. C., Stagg, C. A., Aiba, K., Hamatani, T., Matoba, R., Kargul, G. J. & Ko, M. S. H. (2003). The NIA cDNA project in mouse stem cells and early embryos. *C R Biol,* **326** (10-11), 931–940.

[18] Celniker, S. E. & Rubin, G. M. (2003). The *Drosophila melanogaster* genome. *Annu Rev Genomics Hum Genet,* **4**, 89–117.

[19] Cherry, J. M., Adler, C., Ball, C., Chervitz, S. A., Dwight, S. S., Hester, E. T., Jia, Y., Juvik, G., Roe, T., Schroeder, M., Weng, S. & Botstein, D. (1998). SGD: Saccharomyces Genome Database. *Nucleic Acids Res,* **26** (1), 73–79.

[20] Claudel-Renard, C., Chevalet, C., Faraut, T. & Kahn, D. (2003). Enzyme-specific profiles for genome annotation: PRIAM. *Nucleic Acids Res,* **31** (22), 6633–6639.

[21] d'Aubenton Carafa, Y., Brody, E. & Thermes, C. (1990). Prediction of rho-independent *Escherichia coli* transcription terminators. A statistical analysis of their RNA stem-loop structures. *J Mol Biol,* **216** (4), 835–858.

[22] Delcher, A. L., Harmon, D., Kasif, S., White, O. & Salzberg, S. L. (1999). Improved microbial gene identification with GLIMMER. *Nucleic Acids Res,* **27** (23), 4636–4641.

[23] Ehrenreich, A. (2006). DNA microarray technology for the microbiologist: an overview. *Appl Microbiol Biotechnol,* **73** (2), 255–273.

[24] Ermolaeva, M. D., Khalak, H. G., White, O., Smith, H. O. & Salzberg, S. L. (2000). Prediction of transcription terminators in bacterial genomes. *J Mol Biol,* **301** (1), 27–33.

[25] Fuglsang, A. (2003). Codon optimizer: a freeware tool for codon optimization. *Protein Expr Purif,* **31** (2), 247–249.

[26] Furey, T. S., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M. & Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics,* **16** (10), 906–914.

[27] Galagan, J. E., Calvo, S. E., Cuomo, C., Ma, L.-J., Wortman, J. R., Batzoglou, S., Lee, S.-I., Baştürkmen, M., Spevak, C. C., Clutterbuck, J., Kapitonov, V., Jurka, J., Scazzocchio, C., Farman, M., Butler, J., Purcell, S., Harris, S., Braus, G. H., Draht, O., Busch, S., D'Enfert, C., Bouchier, C., Goldman, G. H., Bell-Pedersen, D., Griffiths-Jones, S., Doonan, J. H., Yu, J., Vienken, K., Pain, A., Freitag, M., Selker, E. U., Archer, D. B., Peñalva, M. A., Oakley, B. R., Momany, M., Tanaka, T., Kumagai, T., Asai, K., Machida, M., Nierman, W. C., Denning, D. W., Caddick, M., Hynes, M., Paoletti, M., Fischer, R., Miller, B., Dyer, P., Sachs, M. S., Osmani, S. A. & Birren, B. W. (2005). Sequencing of *Aspergillus*

nidulans and comparative analysis with *A. fumigatus* and *A. oryzae*. *Nature,* **438** (7071), 1105–1115.

[28] Gao, W., Rzewski, A., Sun, H., Robbins, P. D. & Gambotto, A. (2004). UpGene: Application of a web-based DNA codon optimization algorithm. *Biotechnol Prog,* **20** (2), 443–448.

[29] Goldman, E., Rosenberg, A. H., Zubay, G. & Studier, F. W. (1995). Consecutive low-usage leucine codons block translation only when near the 5' end of a message in *Escherichia coli*. *J Mol Biol,* **245** (5), 467–473.

[30] Gouy, M. & Gautier, C. (1982). Codon usage in bacteria: correlation with gene expressivity. *Nucleic Acids Res,* **10** (22), 7055–7074.

[31] Gustafsson, C., Govindarajan, S. & Minshull, J. (2004). Codon bias and heterologous protein expression. *Trends Biotechnol,* **22** (7), 346–353.

[32] Hempel, D. (2006). Integration gen- und verfahrenstechnischer Methoden zur Entwicklung biotechnologischer Prozesse Sonderforschungsbereich 578 - Vom Gen zum Produkt. *Chemie Ingenieur Technik,* **78** (3), 187–192.

[33] Hiller, K., Grote, A., Maneck, M., Münch, R. & Jahn, D. (2006). JVirGel 2.0: computational prediction of proteomes separated via two-dimensional gel electrophoresis under consideration of membrane and secreted proteins. *Bioinformatics,* **22** (19), 2441–2443.

[34] Hoover, D. M. & Lubkowski, J. (2002). DNAWorks: an automated method for designing oligonucleotides for PCR-based gene synthesis. *Nucleic Acids Res,* **30** (10), e43.

[35] Hundertmark, C. (2005). Entwicklung und Verwendung eines datenbankgestützten Webportals zur bioinformatischen Genomannotation von *Bacillus megaterium*. Diplomarbeit, Institut für Mikrobiologie, Technische Universität Braunschweig.

[36] Ikemura, T. (1981). Correlation between the abundance of *Escherichia coli* transfer RNAs and the occurrence of the respective codons in its protein genes. *J Mol Biol,* **146** (1), 1–21.

[37] Ikemura, T. (1985). Codon usage and tRNA content in unicellular and multicellular organisms. *Mol Biol Evol,* **2** (1), 13–34.

[38] Imanishi, T., Itoh, T., Suzuki, Y., O'Donovan, C., Fukuchi, S., Koyanagi, K. O., Barrero, R. A., Tamura, T., Yamaguchi-Kabata, Y., Tanino, M., Yura, K., Miyazaki, S., Ikeo, K., Homma, K., Kasprzyk, A., Nishikawa, T., Hirakawa, M., Thierry-Mieg, J., Thierry-Mieg, D., Ashurst, J., Jia, L., Nakao, M., Thomas, M. A., Mulder, N., Karavidopoulou, Y., Jin, L., Kim, S., Yasuda, T., Lenhard, B., Eveno, E., Suzuki, Y., Yamasaki, C., ichi Takeda, J., Gough, C., Hilton, P., Fujii, Y., Sakai, H., Tanaka, S., Amid, C., Bellgard, M., de Fatima Bonaldo, M., Bono, H., Bromberg, S. K., Brookes, A. J., Bruford, E., Carninci, P., Chelala, C., Couillault, C., de Souza, S. J., Debily, M.-A., Devignes, M.-D., Dubchak, I., Endo, T., Estreicher, A., Eyras, E., Fukami-Kobayashi, K., Gopinath, G. R., Graudens, E., Hahn, Y., Han, M., Han, Z.-G., Hanada, K., Hanaoka, H., Harada, E., Hashimoto, K., Hinz, U., Hirai, M., Hishiki, T., Hopkinson, I., Imbeaud, S., Inoko, H., Kanapin, A., Kaneko, Y., Kasukawa, T., Kelso, J., Kersey, P., Kikuno, R., Kimura, K., Korn, B., Kuryshev, V., Makalowska, I., Makino, T., Mano, S., Mariage-Samson, R., Mashima, J., Matsuda, H., Mewes, H.-W., Minoshima, S., Nagai, K., Nagasaki, H., Nagata, N., Nigam, R., Ogasawara, O., Ohara, O., Ohtsubo, M., Okada, N., Okido, T., Oota, S., Ota, M., Ota, T., Otsuki, T., Piatier-Tonneau, D., Poustka, A., Ren, S.-X., Saitou, N., Sakai, K., Sakamoto, S., Sakate, R., Schupp, I., Servant, F., Sherry, S., Shiba, R., Shimizu, N., Shimoyama, M., Simpson, A. J., Soares, B., Steward, C., Suwa, M., Suzuki, M., Takahashi, A., Tamiya, G., Tanaka, H., Taylor, T., Terwilliger, J. D., Unneberg, P., Veeramachaneni, V., Watanabe, S., Wilming, L., Yasuda, N., Yoo, H.-S., Stodolsky, M., Makalowski, W., Go, M., Nakai, K., Takagi, T., Kanehisa, M., Sakaki, Y., Quackenbush, J., Okazaki, Y., Hayashizaki, Y., Hide, W., Chakraborty, R., Nishikawa, K., Sugawara, H., Tateno, Y., Chen, Z., Oishi, M., Tonellato, P., Apweiler, R., Okubo, K., Wagner, L., Wiemann, S., Strausberg, R. L., Isogai, T., Auffray, C., Nomura, N., Gojobori, T. & Sugano, S. (2004). Integrative annotation of 21,037 human genes validated by full-length cDNA clones. *PLoS Biol,* **2** (6), e162.

[39] Jayaraj, S., Reid, R. & Santi, D. V. (2005). GeMS: an advanced software package for designing synthetic genes. *Nucleic Acids Res,* **33** (9), 3011–3016.

[40] Kalate, R. N., Tambe, S. S. & Kulkarni, B. D. (2003). Artificial neural networks for prediction of mycobacterial promoter sequences. *Comput Biol Chem,* **27** (6), 555–564.

[41] Kanehisa, M., Goto, S., Hattori, M., Aoki-Kinoshita, K. F., Itoh, M., Kawashima, S., Katayama, T., Araki, M. & Hirakawa, M. (2006). From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res,* **34** (Database issue), D354–D357.

[42] Kauffman, K. J., Prakash, P. & Edwards, J. S. (2003). Advances in flux balance analysis. *Curr Opin Biotechnol,* **14** (5), 491–496.

[43] Kell, D. B. (2004). Metabolomics and systems biology: making sense of the soup. *Curr Opin Microbiol,* **7** (3), 296–307.

[44] Kersey, P., Bower, L., Morris, L., Horne, A., Petryszak, R., Kanz, C., Kanapin, A., Das, U., Michoud, K., Phan, I., Gattiker, A., Kulikova, T., Faruque, N., Duggan, K., Mclaren, P., Reimholz, B., Duret, L., Penel, S., Reuter, I. & Apweiler, R. (2005). Integr8 and Genome Reviews: integrated views of complete genomes and proteomes. *Nucleic Acids Res,* **33** (Database issue), D297–D302.

[45] Kirschner, M. W. (2005). The meaning of systems biology. *Cell,* **121** (4), 503–504.

[46] Kobayashi, T., Abe, K., Asai, K., Gomi, K., Juvvadi, P. R., Kato, M., Kitamoto, K., Takeuchi, M. & Machida, M. (2007). Genomics of *Aspergillus oryzae*. *Biosci Biotechnol Biochem,* **71** (3), 646–670.

[47] Lansing M. Prescott, John P. Harley, D. A. K. (1999). *Microbiology*. 4th edition edition, WCB McGraw-Hill.

[48] Lavner, Y. & Kotlar, D. (2005). Codon bias as a factor in regulating expression via translation rate in the human genome. *Gene,* **345** (1), 127–138.

[49] Lipman, D. J. & Pearson, W. R. (1985). Rapid and sensitive protein similarity searches. *Science,* **227** (4693), 1435–1441.

[50] Lowe, T. M. & Eddy, S. R. (1997). tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic Acids Res,* **25** (5), 955–964.

[51] Machida, M., Asai, K., Sano, M., Tanaka, T., Kumagai, T., Terai, G., Kusumoto, K.-I., Arima, T., Akita, O., Kashiwagi, Y., Abe, K., Gomi, K., Horiuchi, H., Kitamoto, K., Kobayashi, T., Takeuchi, M., Denning, D. W., Galagan, J. E., Nierman, W. C., Yu, J., Archer, D. B., Bennett, J. W., Bhatnagar, D., Cleveland, T. E., Fedorova, N. D., Gotoh, O., Horikawa, H., Hosoyama, A., Ichinomiya, M., Igarashi, R., Iwashita, K., Juvvadi, P. R., Kato, M., Kato, Y., Kin, T., Kokubun, A., Maeda, H., Maeyama, N., ichi Maruyama, J., Nagasaki, H., Nakajima, T., Oda, K., Okada, K., Paulsen, I., Sakamoto, K., Sawano, T., Takahashi, M., Takase, K., Terabayashi, Y., Wortman, J. R., Yamada, O., Yamagata, Y., Anazawa, H., Hata, Y., Koide, Y., Komori, T., Koyama, Y., Minetoki, T., Suharnan, S., Tanaka, A., Isono, K., Kuhara, S., Ogasawara, N. & Kikuchi, H. (2005). Genome sequencing and analysis of *Aspergillus oryzae*. *Nature,* **438** (7071), 1157–1161.

[52] Majoros, W. H., Pertea, M., Antonescu, C. & Salzberg, S. L. (2003). GlimmerM, Exonomy and Unveil: three ab initio eukaryotic genefinders. *Nucleic Acids Res,* **31** (13), 3601–3604.

[53] Majoros, W. H., Pertea, M. & Salzberg, S. L. (2004). TigrScan and GlimmerHMM: two open source ab initio eukaryotic gene-finders. *Bioinformatics,* **20** (16), 2878–2879.

[54] Materi, W. & Wishart, D. S. (2007). Computational systems biology in drug discovery and development: methods and applications. *Drug Discov Today,* **12** (7-8), 295–303.

[55] Melzer, G., Dalpiaz, A., Grote, A., Kucklick, M., Göcke, Y., Jonas, R., Dersch, P., Franco-Lara, E., Nörtemann, B. & Hempel, D. C. (2007). Metabolic flux analysis using stoichiometric models for *Aspergillus niger*: Comparison under glucoamylase-producing and non-producing conditions. *J Biotechnol,* **132** (4), 405–417.

[56] Michael T. Madigan, John M. Martinko, J. P. (2000). *Brock Mikrobiologie*. Spektrum Akademischer Verlag Heidelberg.

[57] Needleman, S. B. & Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol,* **48** (3), 443–453.

[58] Nierman, W. C., Pain, A., Anderson, M. J., Wortman, J. R., Kim, H. S., Arroyo, J., Berriman, M., Abe, K., Archer, D. B., Bermejo, C., Bennett, J., Bowyer, P., Chen, D., Collins, M., Coulsen, R., Davies, R., Dyer, P. S., Farman, M., Fedorova, N., Fedorova, N., Feldblyum, T. V., Fischer, R., Fosker, N., Fraser, A., García, J. L., García, M. J., Goble, A., Goldman, G. H., Gomi, K., Griffith-Jones, S., Gwilliam, R., Haas, B., Haas, H., Harris, D., Horiuchi, H., Huang, J., Humphray, S., Jiménez, J., Keller, N., Khouri, H., Kitamoto, K., Kobayashi, T., Konzack, S., Kulkarni, R., Kumagai, T., Lafon, A., Lafton, A., Latgé, J.-P., Li, W., Lord, A., Lu, C., Majoros, W. H., May, G. S., Miller, B. L., Mohamoud, Y., Molina, M., Monod, M., Mouyna, I., Mulligan, S., Murphy, L., O'Neil, S., Paulsen, I., Peñalva, M. A., Pertea, M., Price, C., Pritchard, B. L., Quail, M. A., Rabbinowitsch, E., Rawlins, N., Rajandream, M.-A., Reichard, U., Renauld, H., Robson, G. D., de Córdoba, S. R., Rodríguez-Peña, J. M., Ronning, C. M., Rutter, S., Salzberg, S. L., Sanchez, M., Sánchez-Ferrero, J. C., Saunders, D., Seeger, K., Squares, R., Squares, S., Takeuchi, M., Tekaia, F., Turner, G., de Aldana, C. R. V., Weidman, J., White, O., Woodward, J., Yu, J.-H., Fraser, C., Galagan, J. E., Asai, K., Machida, M., Hall, N., Barrell, B. & Denning, D. W. (2005). Genomic sequence of the pathogenic and allergenic filamentous fungus *Aspergillus fumigatus*. *Nature,* **438** (7071), 1151–1156.

[59] Pel, H. J., de Winde, J. H., Archer, D. B., Dyer, P. S., Hofmann, G., Schaap, P. J., Turner, G., de Vries, R. P., Albang, R., Albermann, K., Andersen, M. R., Bendtsen, J. D., Benen, J. A. E., van den Berg, M., Breestraat, S., Caddick, M. X., Contreras, R., Cornell, M., Coutinho, P. M., Danchin, E. G. J., Debets, A. J. M., Dekker, P., van Dijck, P. W. M., van Dijk, A., Dijkhuizen, L., Driessen, A. J. M., d'Enfert, C., Geysens, S., Goosen, C., Groot, G. S. P., de Groot, P. W. J., Guillemette, T., Henrissat, B., Herweijer, M., van den Hombergh, J. P. T. W., van den

Hondel, C. A. M. J. J., van der Heijden, R. T. J. M., van der Kaaij, R. M., Klis, F. M., Kools, H. J., Kubicek, C. P., van Kuyk, P. A., Lauber, J., Lu, X., van der Maarel, M. J. E. C., Meulenberg, R., Menke, H., Mortimer, M. A., Nielsen, J., Oliver, S. G., Olsthoorn, M., Pal, K., van Peij, N. N. M. E., Ram, A. F. J., Rinas, U., Roubos, J. A., Sagt, C. M. J., Schmoll, M., Sun, J., Ussery, D., Varga, J., Vervecken, W., van de Vondervoort, P. J. J., Wedler, H., Wösten, H. A. B., Zeng, A.-P., van Ooyen, A. J. J., Visser, J. & Stam, H. (2007). Genome sequencing and analysis of the versatile cell factory *Aspergillus niger* CBS 513.88. *Nat Biotechnol,* **25** (2), 221–231.

[60] Pinney, J. W., Shirley, M. W., McConkey, G. A. & Westhead, D. R. (2005). metaSHARK: software for automated metabolic network prediction from DNA sequence and its application to the genomes of *Plasmodium falciparum* and *Eimeria tenella*. *Nucleic Acids Res,* **33** (4), 1399–1409.

[61] Puigbò, P., Guzmán, E., Romeu, A. & Garcia-Vallvé, S. (2007). OPTIMIZER: a web server for optimizing the codon usage of DNA sequences. *Nucleic Acids Res,* **35** (Web Server issue), W126–W131.

[62] Richardson, S. M., Wheelan, S. J., Yarrington, R. M. & Boeke, J. D. (2006). GeneDesign: rapid, automated design of multikilobase synthetic genes. *Genome Res,* **16** (4), 550–556.

[63] Roberts, R. J., Vincze, T., Posfai, J. & Macelis, D. (2003). REBASE: restriction enzymes and methyltransferases. *Nucleic Acids Res,* **31** (1), 418–420.

[64] Roberts, R. J., Vincze, T., Posfai, J. & Macelis, D. (2007). REBASE–enzymes and genes for DNA restriction and modification. *Nucleic Acids Res,* **35** (Database issue), D269–D270.

[65] Salzberg, S. L., Pertea, M., Delcher, A. L., Gardner, M. J. & Tettelin, H. (1999). Interpolated Markov models for eukaryotic gene finding. *Genomics,* **59** (1), 24–31.

[66] Schlegel, H.-G. (1992). *Allgemeine Mikrobiologie*. Georg Thieme Verlag Stuttgart.

[67] Sharp, P. M. & Li, W. H. (1986). Codon usage in regulatory genes in *Escherichia coli* does not reflect selection for 'rare' codons. *Nucleic Acids Res,* **14** (19), 7737–7749.

[68] Sharp, P. M. & Li, W. H. (1987). The codon Adaptation Index–a measure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Res,* **15** (3), 1281–1295.

[69] Singleton, P. & Sainsbury, D. (1978). *Dictionary of Microbiology*. John Wiley & Sons.

[70] Smith, T. F. & Waterman, M. S. (1981). Identification of common molecular subsequences. *J Mol Biol,* **147** (1), 195–197.

[71] Sterk, P., Kersey, P. J. & Apweiler, R. (2006). Genome Reviews: standardizing content and representation of information about complete genomes. *OMICS,* **10** (2), 114–118.

[72] Sørensen, M. A., Kurland, C. G. & Pedersen, S. (1989). Codon usage determines translation rate in *Escherichia coli*. *J Mol Biol,* **207** (2), 365–377.

[73] Vary, P. S., Biedendieck, R., Fuerch, T., Meinhardt, F., Rohde, M., Deckwer, W.-D. & Jahn, D. (2007). *Bacillus megaterium*–from simple soil bacterium to industrial protein production host. *Appl Microbiol Biotechnol,* **76** (5), 957–967.

[74] Villalobos, A., Ness, J. E., Gustafsson, C., Minshull, J. & Govindarajan, S. (2006). Gene Designer: a synthetic biology tool for constructing artificial DNA segments. *BMC Bioinformatics,* **7**, 285.

[75] Vogelbacher, R., Angulo, D. & Koide, S. (2006). Sequence optimization for synthetic genes using a genetic algorithm. *Proceedings of the Midwest Software Engineering Conference/DePaul CTI Research Symposium,* **1**.

[76] Wu, G., Bashir-Bello, N. & Freeland, S. J. (2006). The Synthetic Gene Designer: a flexible web platform to explore sequence manipulation for heterologous expression. *Protein Expr Purif,* **47** (2), 441–445.

[77] Wu, G., Dress, L. & Freeland, S. J. (2007*a*). Optimal encoding rules for synthetic genes: the need for a community effort. *Mol Syst Biol,* **3**, 134.

[78] Wu, G., Zheng, Y., Qureshi, I., Zin, H. T., Beck, T., Bulka, B. & Freeland, S. J. (2007*b*). SGDB: a database of synthetic genes re-designed for optimizing protein over-expression. *Nucleic Acids Res,* **35** (Database issue), D76–D79.

[79] Yang, Y. (2007). *Production and secretion of recombinant proteins using Bacillus megaterium*. Doktorarbeit, Technische Universität Braunschweig.

[80] Yang, Y., Malten, M., Grote, A., Jahn, D. & Deckwer, W.-D. (2007). Codon optimized *Thermobifida fusca* hydrolase secreted by *Bacillus megaterium*. *Biotechnol Bioeng,* **96** (4), 780–794.